Septiembre 1985 250 ptas.

TodoSpectrum

AÑO 2 - NUMERO 13. REVISTA EXCLUSIVA PARA USUARIOS

Periféricos DISCOVERY 1

Listas escolares con Spectrum

Variaciones sobre el Sistema Operativo

Programa especial: PARCHIS



AUTOMATA U.K. PRESENTS
YOURSELF
IAN DURY
JON PERTWEE
MEL CROUCHER
DONNA BAILEY
and FRANKIE HOWERD in
DEVS EX MACHINA
WRITTEN AND DIRECTED BY MEL CROUCHER

SPECTRUM
DEUS EX MACHINA. ¡EL AUDIO-VIDEO!

Un nuevo concepto de juego por ordenador llega de la mano de Investrónica.

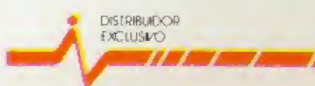
Deus ex Machina.

Una historia de Ciencia-Ficción creada por Andrew Stagg, con música de Mel Croucher.

Siéntate ante tu televisor ... sincroniza la banda sonora y sumérgete en un espectáculo total.

Ha nacido el audio-video por ordenador.

SPECTRUM. EL MAXIMUN EN SOFTWARE



Tomás Bretón 60 Telf (91) 467 82 10 Télex 23399 IYCO E 28045 Madrid
Camp. 80 Telf (93) 211 26 58-211 27 54 08022 Barcelona



Portada: Puchecha

Director: Simeón Cruz

13

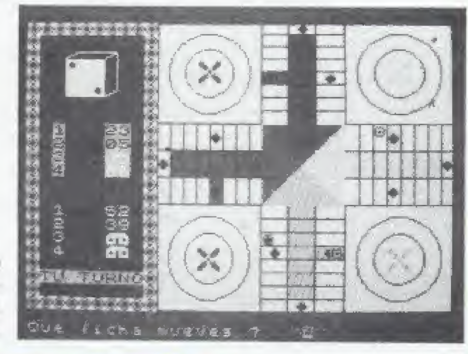
AÑO II



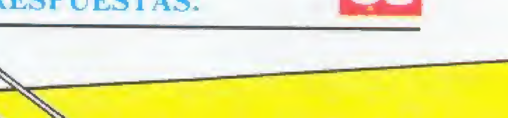
El Discovery y el Spectrum, una pareja muy bien avenida.



Inteligencia artificial: un buen Interface hombre-máquina.



El Parchís, un juego de ordenador sin violencia y sin marcianos.



Pasadas las vacaciones estivales, volvemos a dedicar las páginas centrales a lo último de Sinclair: el QL. Ordenador que gana adeptos día a día, aunque aún le queda mucho para alcanzar las cotas de su hermano menor el Spectrum.

Un Spectrum todo-terreno, que sirve, como vemos en este número, tanto para jugar al parchís como para ayudar a los profesores en la pesada labor de todos los cursos: ordenar e imprimir la lista de los nuevos alumnos.

ACTUALIDAD

10 REM NOS VEMOS EN EL SONIMAG. Como todos los años, este mes de Septiembre se celebra el Sonimag, que incorpora una sección de microinformática junto a las ya tradicionales de imagen y sonido. Pese a la saturación que está comenzando a alcanzar el mercado, se esperan novedades para Spectrum, sobre todo en materia de software. También estará el QL, que puede responder bien a la presión de Amstrad y MSX, con la presentación de programas profesionales y la unidad de disco "oficial" Sinclair, que distribuye Investrónica.

20 REM ¿QUIEN NO CONOCE EL Tasword? Este procesador de texto para Spectrum se ha convertido en un auténtico Best Seller. Para los que usan la versión española, Context, comercializada por Pin-Soft, una buena noticia: acaba de aparecer la versión ocho, que permite, entre otras mejoras, utilizar los caracteres especiales del catalán. Y es que la autonomía se nota en todos los ámbitos.

30 REM Y SINCLAIR SIGUE DANDO QUE HABLAR. Ahora surgen rumores de dificultades en las negociaciones con Maxwell; dificultades que pueden retrasar la carrera tecnológica del laboratorio de Cambridge, en un momento en que entramos ya en la carrera de las ventas navideñas. Los rumores apuntan hacia el Pandora, un compatible Spectrum totalmente portátil con 128K RAM. Sir Clive guarda un prudente silencio.

40 REM EL CENTRO NACIONAL DE MICROELECTRONICA echó a andar este verano. Dependiente del Consejo Superior de Investigaciones Científicas, intenta potenciar la investigación y desarrollo de una tecnología española en este sector. Le deseamos larga y, sobre todo, fructífera vida.



DISCOVERY 1

Unidad de disco

Dado el gran número de alternativas para el almacenamiento masivo en el Spectrum, parece que la aparición de una unidad de disco sólo debe justificarse si aporta algo realmente nuevo. El Discovery *nos ofrece una unidad clásica y sin experimentos, pero muy válida como alternativa a los restantes sistemas.*

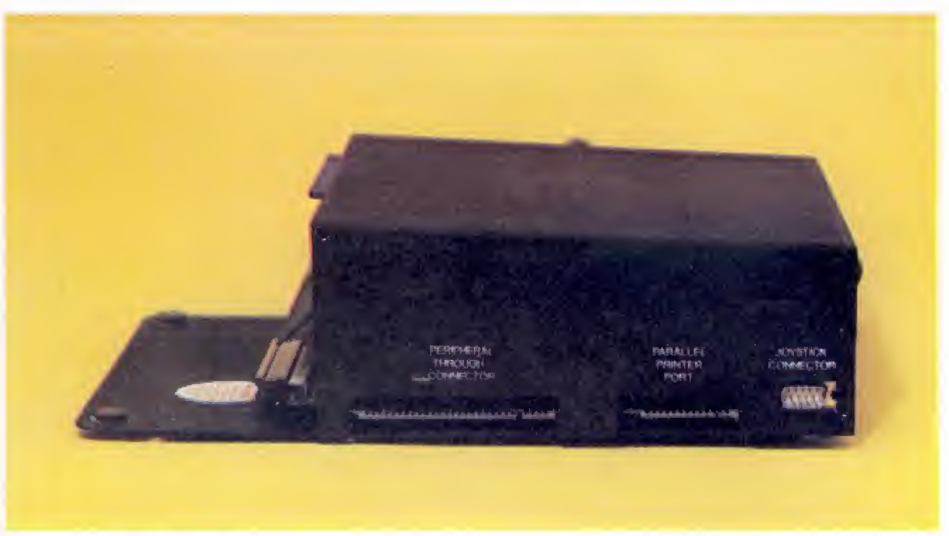
Una unidad de disco bien adaptada al Spectrum

Una de las principales virtudes del *Discovery* es que ha sido diseñado pensando en el Spectrum. En efecto, una vez conectado, y usando el tornillo y los dos topes de goma que se nos facilitan con la unidad, el conjunto queda firme-

mente conectado, ofreciendo una gran comodidad de acceso a las funciones del Spectrum.

La unidad presenta un ancho algo inferior al del Spectrum. Su base presenta una prolongación en la parte anterior, con un *bus* de expansión que permite una buena fijación del ordenador. El único problema aparente es que el *Discovery* «aprisiona» la parte trasera del Spectrum, con la consiguiente dificultad para conectar la salida de TV y la toma de *cassette*. El conector de alimentación no se puede conectar, pero eso no es un inconveniente, ya que el *Discovery* proporciona la corriente al sistema entero. ¡Menos cables en nuestra mesa!

Nosotros no hemos tenido ningún problema en la «maniobra de acoplamiento». Más problemático es extraer el cable de *cassette* o TV, ya que habrá que destornillar el ordenador. Quienes dispongan de monitor de vídeo compuesto no sufrirán este inconveniente, ya que pueden conectar su aparato a una salida de vídeo que hay en la parte trasera de la unidad.



Conexiones del Discovery.

Igual que otras unidades, se añaden algunos extras junto con la unidad de disco. *Discovery 1* incluye un *port Centronics* y una conexión para *joystick* tipo *Kempston*. Siendo éstos los periféricos más vendidos para Spectrum, parece lógica su inclusión. El lateral derecho del aparato incluye estas dos conexiones, junto con una prolongación del *bus* del Spectrum.

El Discovery por los caminos de Santiago

Para Juan Rodríguez el **Discovery** es un auténtico descubrimiento. Lleva la dirección de **Halt Software**, empresa afincada en Santiago de Compostela, canalizando la actividad informática que ofrece una ciudad eminentemente estudiantil como ésta.

A principios de junio se decidieron por el **Discovery**. Ahora, transcurrido un tiempo prudencial de estudio del equipo, nos cuenta sus experiencias.

—¿Por qué una unidad de discos para el Spectrum?

—Todo empezó en el momento en que quise realizar programas un poco serios. El *cassette* me producía múltiples trastornos psicosomáticos. El *microdrive* me gustaba, pero no resolvía mis problemas: al utilizarlo asiduamente para programar la fiabilidad, aunque superior al *cassette*, resultaba insuficiente. Podían fallar muchas cosas: el Spectrum, la corriente, yo mismo. Pero perder ocasionalmente las 90 K de un cartucho me sumió en profundas depresiones periódicas de difícil superación...

—¿Supongo que ahí es cuando entra en juego el **Discovery**?

—Sí, me interesé por las distintas unidades del mercado. Estuve a punto de comprar la de Timex **y la** BETA. **Pero al final me decidí por** el Discovery.

—¿Y fue una decisión positiva?

—Por supuesto. Los ficheros de acceso directo evocaron en mí viejos proyectos sobre inteligencia artificial. Pero siguen siendo proyectos. Ahora me dedico a la gestión de pequeños negocios. Lo del disco de RAM es un buen invento, consistente en manejar 32 K de RAM como si fuese la quinta unidad de disco. Las opciones adicionales hacen posible una gestión de pequeños negocios digna con una inversión reducida. El Discovery **no sólo sirve para programar más cómodamente, también sirve para programar el** Discovery. **Interesante ¿verdad?**

—¿Y en cuanto a los defectos?

—Es difícil esa pregunta: deci los defectos de una señora estupenda es poco elegante porque, en rea- lidad, todos deseamos perdonárse-

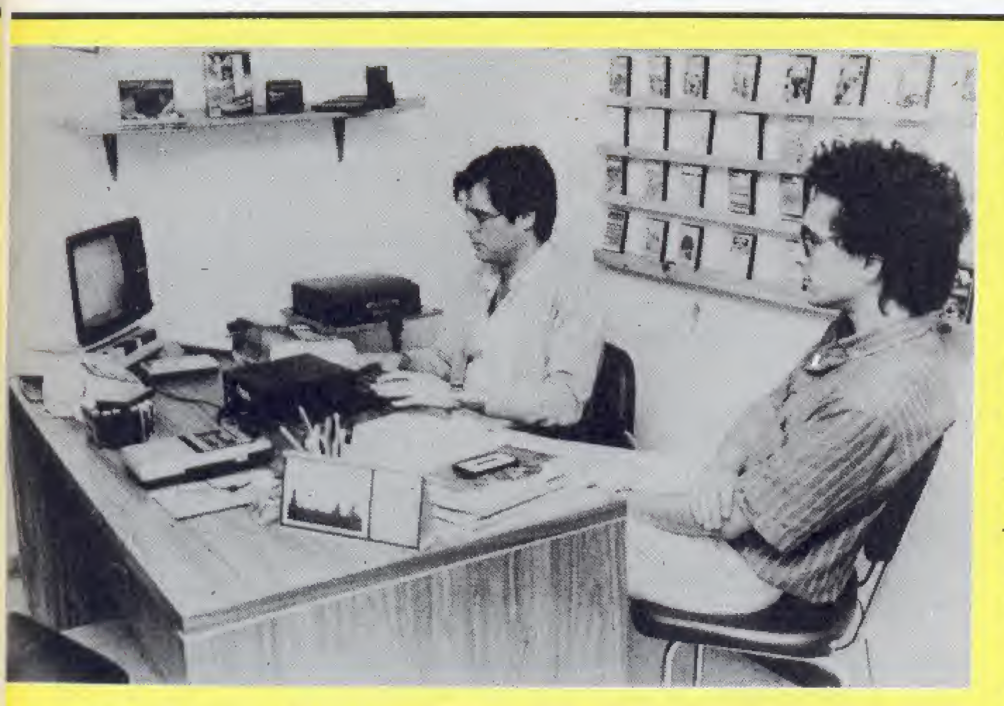
Los discos de 3 1/2 pulgadas se imponen como un nuevo standard



La unidad de disco asoma por encima del ordenador. Los discos son de simple cara/simple densidad.

Pero vamos a lo esencial: la unidad de disco. Esta asoma por encima de nuestro ordenador, en una posición cómoda para su uso. El *Discovery* está disponible en dos versiones, con una o dos unidades de disco de 3 1/2" (formato *Sony*). Los discos son de simple cara/simple densidad, lo que nos da una capacidad aproximada de 180K por *diskette.* Es posible la ampliación de uno a dos lectores, a un precio de alrededor de 30.000 pesetas. Uno de los aspectos más importantes de cualquier disco es el sistema operativo, que complementa el BASIC con instrucciones que permiten el manejo de los nuevos periféricos. En el caso del *Discovery* se ha optado por mantener la compatibilidad con el *microdrive* al máximo, respetándose la sintaxis de los comandos en tanto sea posible. La elección parece lógica, ya que el *microdrive* es un medio de almacenamiento muy extendido, y el que cuenta con más *software* en el mercado.

El sistema de paginación es muy similar al del *Interface I,* pero incluye algunos cambios curiosos. Por ejemplo, si tecleamos PRINT USR 8 el sistema nos contestará imprimiendo el número de la versión que estemos utilizando. A nosotros la prueba nos dio como resultado 2.1. Por otro lado, un USR



los. De todos modos, tengo que confesar que tiene sus sorpresas.

En primer lugar, viene con un enchufe alucinante, pero totalmente inútil para la instalación española. Parece querer decir que los electrones anglosajones son más «chachis» que los tercermundistas.

Realizado el cambio, si algo no funciona son cosas de la ULA.

Otra sorpresa es el disco de tres pulgadas y media. La información es insuficiente, imprecisa, trivial y a veces errónea.

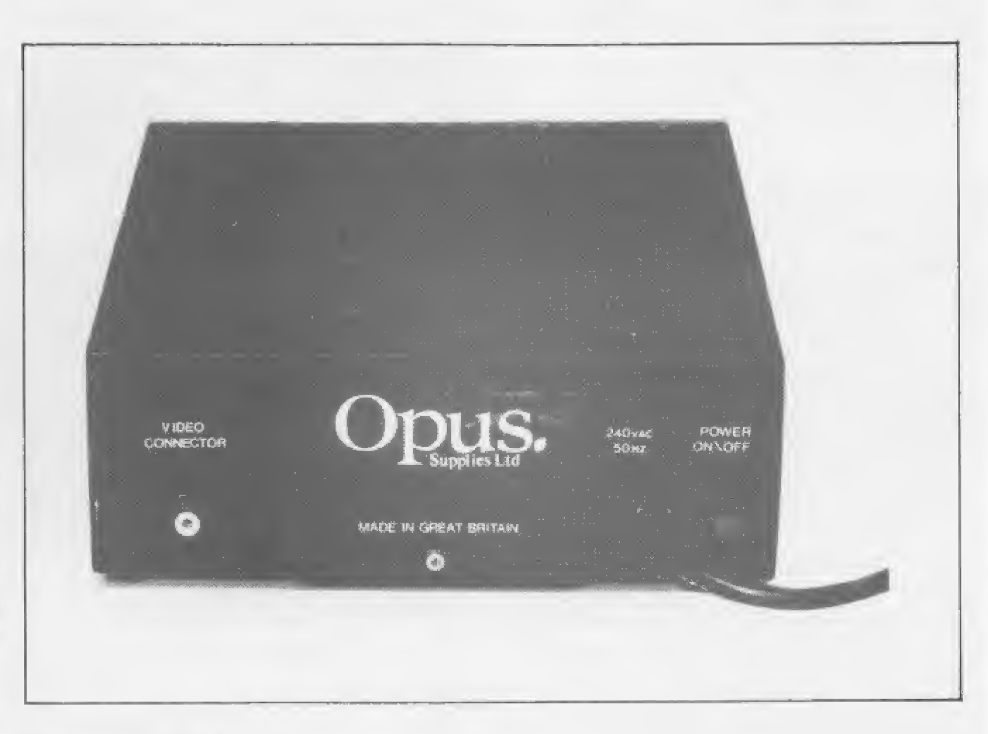
Hay quien dice que otro problema es que no hay programas específicos para el Discovery. **Yo creo que esto es bueno. Siempre es posible hacerlos o adaptarlos, y eso significa trabajo para quien sepa hacerlo.**

—Finalmente, ¿a quién se la recomendarías?

—El asunto tiene dos vertientes: el programador y el usuario. Desde el punto de vista del programador no se puede trabajar con *cassette* **sin perder los nervios. Si quieres desarrollar juegos o programas orientados al** *cassette,* **necesitas el** *microdrive.* **Si los programas están orientados al** *microdrive,* **es recomendable el** Discovery.

Desde el punto de vista del usuario el *cassette* **es suficiente para juegos. Si utiliza el Spectrum para la pequeña gestión, debería bastarle el** *microdrive.* **Si no le llega, el** Discovery **es lo único razonable. Pero no cabe duda que para este usuario, la elección del equipo depende de la disponibilidad de** *software* **adecuado a sus necesidades específicas.**

14070 nos hará RESET del sistema de disco y el ordenador. USR 0 sólo pone a cero el Spectrum, sin afectar al sistema de disco.



Vista posterior del Discovery.

Los comandos son compatibles con los del Interface I

El primer paso para utilizar un disco es formatearlo. Para ello basta teclear FORMAT "m";n;"Nombre". Después de unos quince segundos, el disco habrá sido preparado para su uso. A partir de ese momento, el comando CAT n nos contesta con una lista encabezada por el nombre del disco, seguida por los nombres de los ficheros existentes, y finalizada por el espacio libre en el disco. Los números de unidades están asignados a las unidades existentes, pudiendo usarse el número 3 para formatear un segundo disco en la unidad 1 y hacer copias en la versión con una sola unidad.

Cuando usemos la unidad 5 el sistema entenderá que nos referimos a un *RAM disk*, es decir, a una simulación de disco que utiliza la memoria de nuestro ordenador. Para utilizarlo hace falta introducir CLEAR 32767, y después una sentencia FORMAT "m";5;"Nombre". A partir de ese momento, podemos salvar y cargar ficheros del disco en RAM, con la salvedad de que toda la información desaparecerá cuando conectemos la alimentación de la unidad. Si bien la idea de un disco en RAM parece ingeniosa, no la encontramos de gran interés en una máquina que dispone de sólo 40K libres, de las que 32 son utilizadas por el disco. Sólo parece útil para trabajos de programación en los que se modifique repetidas veces un programa pequeño. En este caso se pueden salvar en el *RAM disk* las distintas versiones y probarlas, con cuidado de no olvidar pasar la información al disco real antes de dar por terminado el trabajo.

Con el *Interface I*, el *Discovery* amplía el sistema de canales que utiliza el Spectrum para comunicarse con los periféricos. Los canales añadidos son: «j», que controla el *joystick*. «t» y «b» se encargan de la comunicación mediante el *Interface* paralelo, y «m», que se refiere a ficheros de disco.

Más canales e instrucciones más potentes

Hasta aquí igual que la expansión de *Sinclair*. Las diferencias empiezan a aparecer con el canal «d», que se refiere al disco entero en operaciones de copia. « #», que se utiliza para asignar un canal a otro canal previamente abierto; «CAT» nos permite leer o escribir directamente del directorio de un disco (aunque la escritura se nos desaconseje, ya que hay peligro de estropear los datos almacenados). «CODE», por último, se utiliza para leer o escribir directamente de la memoria, y nos proporciona acceso al *hardware* de la unidad.

Una especificación de canales muy completa, que aporta novedades interesantes sin perder la compatibilidad. Los canales se asignan con la sentencia OPEN # , que también resulta ampliada con respecto al *microdrive*. El nuevo formato es OPEN # flujo;canal ACCESO.

La sintaxis de la instrucción queda ampliada y simplificada, ya que con el *Discovery* no es necesario especificar «m» si el canal se refiere al disco. Por otro lado, el parámetro de acceso permite las siguientes posibilidades: IN, que indica que el fichero sólo se puede leer, y OUT, para los ficheros de escritura.

En el caso del canal «m», se le pueden indicar más opciones: EXP, para EXPandir un fichero secuencial, nos permite escribir a partir del final del fichero; se le puede añadir una longitud, que indica la ampliación máxima que puede sufrir el fichero. También aparece la opción RND n1;n2 que permite crear ficheros de acceso aleatorio. El primer número indica la longitud de cada ficha, y el segundo el número de éstas. Para abrir ficheros ya existentes, basta utilizarla omitiendo el segundo número.

En relación con los ficheros de acceso aleatorio, existe un nuevo comando necesario para posicionarnos antes de leer o escribir: se trata de POINT # flujo;núm., que

posiciona el flujo de que se trate al principio de la ficha indicada. Si se abre un fichero secuencial con acceso aleatorio y longitud de ficha 1, se puede utilizar POINT para recorrerlo con total libertad.

Respecto a los ficheros secuenciales, los únicos de que dispone el *microdrive*, la principal diferencia es que pueden abrirse para entrada o salida sin depender de que existieran o no previamente. También el comportamiento mediante INKEY$ es distinto. El *Discovery* no da error de fin de fichero cuando leemos más allá del fin de un fichero. Simplemente devuelve una cadena vacía. Si deseamos comprobar el fin de fichero, debemos utilizar la instrucción USR 432, que devuelve el número de *bytes* que nos quedan por leer en el flujo «activo». Para convertir un flujo determinado en el activo basta introducir antes PRINT #n;

Los comandos SAVE, LOAD, VERIFY y MERGE trabajan de la forma habitual, aunque en el caso de SAVE el sistema no da un mensaje de error si el fichero ya existe, sustituyéndolo simplemente por la nueva versión. Esta característica, cómoda en muchas ocasiones, resulta peligrosa si no tomamos la precaución de comprobar los nombres en el disco.

Otro comando de mucha importancia es MOVE, que permite transmitir el contenido de un flujo a otro. Una mejora importante con respecto al *microdrive* es que, si hacemos MOVE «k» to «s», por ejemplo, podemos finalizar pulsando SHIFT ENTER. Esto nos permite llenar ficheros directamente desde el teclado.

Abrir directorios, un lujo al alcance de pocos sistemas de disco

Para abrir el directorio de un *diskette* basta hacer OPEN # n-;"CAT". A partir de ese momento podemos leer las diversas entradas, que consisten en 16 *bytes* para cada fichero, con tres punteros y el nombre del archivo al que están asociados. La primera entrada se refiere al propio directorio. Cuando se formatea un disco, el sistema deja sitio para 110 ficheros.

Si el disco contiene muchos ficheros pequeños, podemos quedarnos sin sitio en el directorio antes de que se acabe el esapcio físico del disco. En ese caso, haciendo OPEN #4;"CAT"EXP long. y después CLEAR #4, el sistema nos ampliará el directorio a la lon-

gitud deseada. La longitud debe especificarse siempre como múltiplo de 16 *bytes*. Por la misma razón, es preferible abrir siempre los directorios como OPEN # 4;"CAT"RND 16, y seleccionar la entrada deseada mediante POINT.

Copia de discos y compactación

Existen otras características que diferencian al *Discovery* del *microdrive*. Entre ellas, lo más interesante es que este sistema permite referenciar un disco entero. Por ejemplo, MOVE "d";1 TO "d";3 permite copiar el contenido íntegro del disco uno al tres. El sistema nos indicará cada cierto tiempo que cambiemos el disco y pulsemos



Discovery abre los ficheros tipo CODE, o los programas como si fueran archivos de datos.

cualquier tecla. El tiempo que tarda en realizarse la copia depende de la memoria libre de que disponga el Spectrum en ese momento, ya que el sistema utiliza como *buffer* la memoria libre. En algunos casos, con el disco totalmente lleno, la operación tardó casi cinco minutos.

Otra operación que puede ahorrar tiempo en discos de datos que contengan muchos archivos es la compactación. Se realiza mediante MOVE "d";1 TO "d";1, y en ella el sistema reorganiza los ficheros, que al irse borrando y escribiendo habían desordenado la estructura del disco, para obtener el máximo rendimiento. La operación es lenta, pero resulta interesante realizarla de vez en cuando en discos muy cargados, sobre los que se escriba a menudo.

Discovery nos deja abrir los ficheros tipo CODE o los programas como si se trataran de archivos de datos, sin dar error en ningún caso.

Conclusión. Una unidad rápida y compacta

Durante el tiempo que la hemos utilizado esta unidad de disco nos

YA ESTÁ A LA VENTA



PIDALA EN SU QUIOSCO HABITUAL O SOLICITELA A:

INFODIS, S.A.

Bravo Murillo, 377 • 28020 Madrid

VIRGIN-I-SPECTRUM

CUPON DE PEDIDO

Ruego me envien la cinta cassette **VIRGIN SOFTWARE** al precio de **525 pts.**

El importe lo abonaré: Con mi tarjeta de crédito ☐ American Express ☐

Visa ☐ Interbank ☐

Contra reembolso ☐ Adjunto cheque ☐

Número de mi tarjeta

Fecha de caducidad

NOMBRE

DIRECCION

CIUDAD

C.P.

PROVINCIA

Sin gastos de envio

ha dado una impresión muy positiva. Al compartir la sintaxis de comandos «oficial» de *Sinclair* tiene acceso a una gran cantidad de *software*. Esta compatibilidad está muy clara en BASIC. En código máquina el manual no nos indica si también se ha respetado, y se echa de menos una lista de los «códigos de enganche» *(hook codes)* del *Discovery*. Por otro lado, los comandos se han ampliado lo suficiente como para poder tener acceso a nuevas características que dejan la potencia del sistema a niveles difíciles de superar. Su velocidad de transferencia es buena, tardando una pantalla en cargar alrededor de cinco segundos.



El manual está bien explicado y traducido al castellano, y a eso no estamos acostumbrados. La unidad simplifica las conexiones del Spectrum al hacerse cargo de la alimentación del conjunto e incorporar un *port* de impresora. En resumen, sólo el precio (unas 63.000 pesetas la unidad simple), y la decisión de elegir un disco de 180K existiendo en el mercado unidades de hasta 720K, pueden deslucir nuestro descubrimiento de este otoño.

¡UN CABLE!
¡CON LO QUE ME HA COSTADO Y SOLO ME DAN UN CABLE!
...¡MIRA ENANO, UN CABLE!
¡¡¡UN CABLE!!!
ME ABURRES, GUS
JCT

SERVICIO DE EJEMPLARES ATRASADOS

Complete su colección de

Todospectrum

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.

Núm. 1 • 250 pts.
Cómo usar el microdrive/Programación Basic/Ampliación Basicare/Rutina despertador/Variables del sistema/Entrada datos mediante máscaras/Protección del software/Sintonice su Spectrum/Programas.

Núm. 2 • 250 pts.
Gráficos profesionales/Desplazamiento pixel a pixel/Utilización de rutinas/Construcción del interface centronics/Programas de utilidad para microdrive/Rutina reset en código máquina/Análisis del editor de textos Tasword/Interfaces para impresoras/Programas.

Núm. 3 • 250 pts.
Novedades sonimag '84/Ampliando el Basic/Programas para ordenar programas/Gráficos con el VU-3D/Lenguaje Forth/Archivos en microdrive/Programación de un interface de impresora/Programas.

Núm. 4 • 250 pts.
De profesión: programador/Consola para el Spectrum/Comparación código máquina-Basic/Análisis programa contabilidad /Calendario/Pascal/Programas.

Núm. 5 • 250 pts.
Floppys para Spectrum/Diseño asistido por ordenador/64 Caracteres por linea/Juego de la vida/Pascal/Así hacemos las portadas/Control de evaluaciones/Programas.

Núm. 6 • 250 pts.
Representación de funciones/Todos los caminos conducen a la ROM/Juegos/Pascal/Construcción de un lápiz óptico/Programas de gestión. El SITI/Logo: tortugas para todos/Interrupciones del Z-80/Programas.





DISPONEMOS DE TAPAS ESPECIALES PARA SUS EJEMPLARES DE

Todospectrum

SIN NECESIDAD DE ENCUADERNACION



(cada tapa es para 6 ejemplares)



Para hacer su pedido, rellene este cupón HOY MISMO y envíelo a:

Todospectrum Bravo Murillo, 377
Tel. 733 96 62 - 28020 MADRID

Ruego me envíen los siguientes ejemplares atrasados de TODOSPECTRUM .... al precio de 250 pts.

Por favor envíenme .... tapas para la encuadernación de mis ejemplares de TODOSPECTRUM, al precio de 600 pts. más gastos de envío.

El importe lo abonaré
☐ POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO ☐ AMERICAN EXPRESS ☐ VISA ☐ INTERBANK

Número de mi tarjeta: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Fecha de caducidad .... Firma

NOMBRE ....

DIRECCION ....

CIUDAD .... C. P. ....

PROVINCIA ....

Juegos

GREMLINS

Erbe
Spectrum 48 K
2.300 ptas.

Las aventuras conversacionales con gráficos son un tipo de juego que tenía problemas para encajar fuera de Gran Gretaña; quizá por problemas idiomáticos, aunque hay quienes dicen que los usuarios ingleses tienen unos gustos diferentes a los continentales. **Gremlins** nos dará una oportunidad de averiguar si el público español disfruta o no con este tipo de juegos.

El juego tiene todos los ingredientes para resultar un éxito: los personajes y las situaciones son conocidos, los gráficos son de calidad, la aventura tiene las dosis de dificultad suficientes para no aburrir, y, lo más importante, los textos han sido traducidos al castellano.

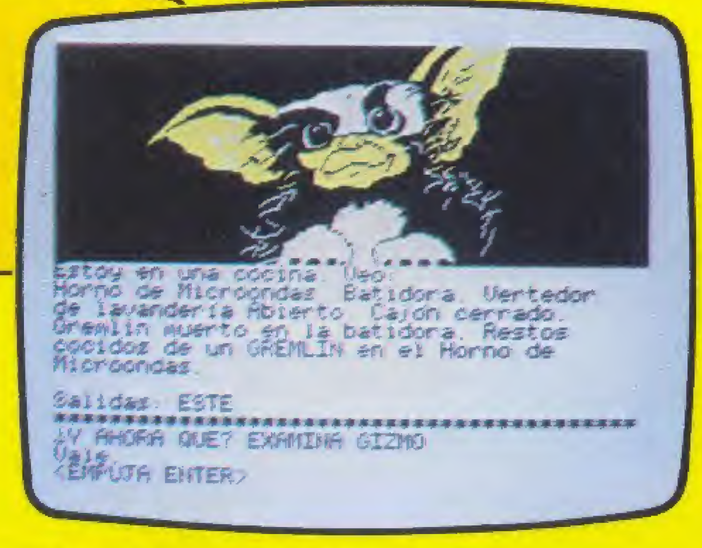
Como quienes hayan visto la película sospecharán, se trata de restablecer el orden en un pueblo plagado de **Gremlins** a los que has dado de comer después de medianoche, engañado por sus trucos. Los retoños de **Gizmo** amenazan con invadir la ciudad, si no consigues acabar con ellos.

Para evitarlo debemos ir matándolos progresivamente, comenzando por nuestra propia casa, y salir a continuación a la calle para completar el trabajo. El comienzo es ya frenético, bastando un pequeño error en el primer movimiento para ser enviados al otro barrio.

Un buen conocimiento de las técnicas y armas requeridas para matar **Gremlins** nos resultará útil, así como recordar las escenas oportunas del film, que nos darán pistas sobre qué hacer en determinadas situaciones.

Como en todos los juegos de aventuras, las dificultades típicas con las que nos tenemos que enfrentar implican pensar rápidamente qué hacer en determinadas circunstancias, aunque más veces de las que sería deseable el problema reside en saber cómo explicarle algo al ordenador. La traducción del juego al castellano ayuda mucho, pero todavía no se puede decir que se haya llegado a hacer programas que «entiendan» nuestro lenguaje.

Otra característica original del juego es que algunas de las imágenes gráficas incluyen movimiento, realzando el realismo de las escenas. Asimismo, la gran velocidad a la que se realizan los cambios de escena impide el aburrimiento. Quizá el punto más débil de



Los «animalitos» no nos dejarán un respiro hasta que hayamos acabado con la plaga.



Si se comete el más ligero error, se paga caro.

Control: Se le dan las órdenes por medio del teclado.
Jugadores: Uno.
Gráficos: Excelentes; una de las mejores aventuras gráficas.
Sonido: No tiene.
Nivel de dificultad: A veces resulta difícil encontrar la manera de salir de una situación.
Originalidad: Se trata de una aventura gráfico-conversacional bastante clásica. Su detalle más original es la incorporación de movimiento en algunos gráficos.
Conclusión: La traducción al castellano de todo el texto de la aventura marca una nueva etapa en la comercialización de juegos de aventuras, que permite el acceso a éstos de muchos usuarios potenciales, impedidos hasta ahora por el idioma.

este juego sea la poca claridad de algunos mensajes, que hacen difícil encontrar la salida en algunas situaciones.

Algunos pequeños fallos de traducción no llegan a empañar la interesante iniciativa de sus autores y distribuidores al traducir el juego. Esperamos que otras compañías de software se decidan a seguir este ejemplo, facilitando el acceso de los usuarios a este tipo de juegos, tan limitado hasta ahora por barreras idiomáticas. Y recuerda: No dejes que les dé el sol, no los mojes y sobre todo,... ¡Nunca juegues después de medianoche!

911 TS

ABC Soft
Spectrum 48 K
1.795 ptas.

Con nuestro *Porsche 911 TS*, y patrocinados por *Dunlop*, vamos a comenzar un duro *Rally* por caminos difíciles, plagados de obstáculos. Antes debemos invertir nuestro dinero (hasta un máximo de 2.000 libras) en la compra de neumáticos, suspensión, etc.

Ya con el preparado, salimos en el primer tramo cronometrado. El coche responde bien, aunque quizá giraría mejor si le hubiésemos puesto una dirección mejor. Unos buenos amortiguadores disminuyen el adaño cuando rozamos un obstáculo...

La carrera se compone de ocho tramos cronometrados, existiendo la posibilidad de recoger equipo (neumáticos, etc.) durante el recorrido de los tramos impares, que son también los que comienzan con opción de compra de material. Para recoger el equipo basta pasar por encima con nuestro coche. Pero hay que estar atentos para no chocar frontalmente contra un árbol, o meternos en un sacavón.

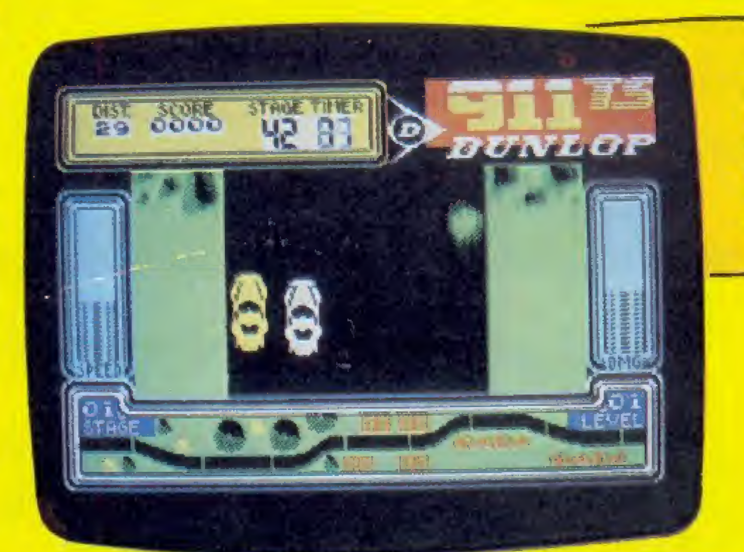
La pantalla tiene una estructura muy similar al *Grand National*, con un pasillo por donde circulamos, y dos laterales muy logrados, donde se nos muestran los daños que ha sufrido el vehículo, la velocidad, y el tiempo que nos falta para finalizar. Cuando acabamos cada tramo, se dan puntos en función del tiempo que nos haya sobrado.

Los golpes que le demos al coche se acumulan, ya que al final de cada tramo el coche se recupera, pero sólo parcialmente. Como consecuencia de esto conviene esquivar los obstáculos cuidadosamente, ya que si los daños son excesivos no podremos continuar la carrera.

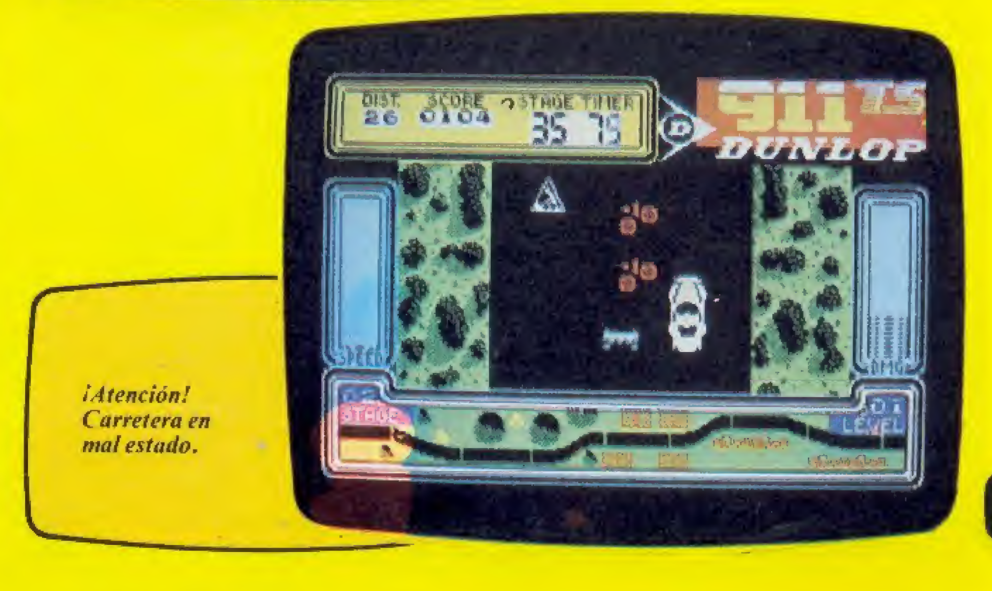
Como principales virtudes del juego podemos citar el scroll, que origina un movimiento muy suave de la pantalla, con un paisaje muy realista en los laterales. También es bueno el movimiento de nuestro coche.

El principal inconveniente es que todo el movimiento del coche se realiza sobre una carretera recta, lo que no contribuye en absoluto a la sensación de velocidad, y esto sería muy necesario en un juego que se supone de acción.

Control: Teclado, Joystick.
Jugadores: Uno.
Gráficos: Buenos, destacando la suavidad del scroll.
Sonido: No es su fuerte.
Nivel de dificultad: Creciente, el comienzo no es excesivamente difícil.
Originalidad: Como todos los juegos de Elite tiene características fuera de lo normal. Incorpora una primera parte en la que debemos comprar accesorios para nuestro automóvil.
Conclusión: Dentro del enfoque clásico de carreras automovilísticas contra el reloj, 911 TS aporta la compra inicial de accesorios para nuestro vehículo.



Ya desde el comienzo hay que moverse rápidamente, si no queremos ser obstaculizados por el otro coche.



¡Atención! Carretera en mal estado.

INTELIGENCIA

En el número anterior comenzamos este artículo presentando al ordenador como un fascinante laboratorio para experimentar las teorías sobre el pensamiento. Hablamos del lenguaje, que es quizá el terreno donde los complejos

La experiencia es una ayuda imprescindible para comprender cualquier fenómeno. Todo lo que nos ocurre se codifica en función de nuestros conocimientos previos del tema. Por esta razón, dos personas pueden vivir una misma situación e interpretarla de distinta forma.

¿Cómo se relaciona esto con los bloques de expectativas de los que hablamos en la primera parte de este artículo? Vimos que los bloques eran un buen método para recordar acontecimientos. En lugar de almacenar cada suceso como una nueva experiencia, un programa puede grabar sólo un puntero con la información no contenida en el prototipo. La memoria sólo necesita registrar las partes de una experiencia que se diferencian del prototipo.

Los bloques de expectativas contienen la información estándar de cada situación, por lo que pueden utilizarse para organizar la memoria. Asumen por tanto un nuevo papel: además de proporcionar las expectativas que ayudan a comprender los acontecimientos, desempeñan la función de memoria.

Para funcionar como memoria y como estructuras de procesamiento tendrán que satisfacer nuevos requisitos. El más importante es que todo sistema de memoria ha de ser dinámico. Nunca será realmente útil si falla al responder a



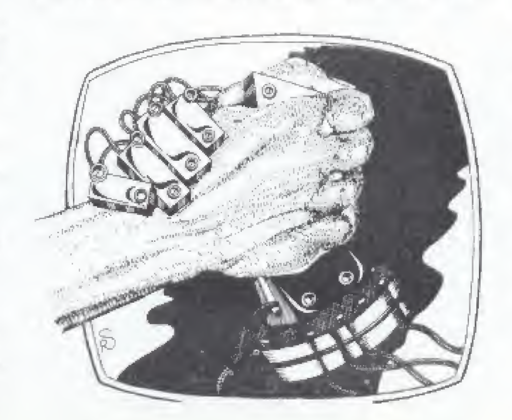
ARTIFICIAL

procesos mentales se expresan con más claridad. Continuamos con el tema donde lo dejamos: empezando a hablar de la memoria, estructura de datos en que se manifiesta lo lejos que estamos de una teoría del pensamiento inteligente.

las nuevas entradas y no aprende de sus experiencias. Si genera expectativa, debe contestar a sus posibles fallos reestructurándose para crear mejores predicciones en el futuro. Además ha de ser capaz de localizar sus conocimientos.

La misión de la memoria es tomar un suceso determinado y encontrar otros relacionados con él. Pero ¿cómo se discrimina entre los que guardan relación y los restantes? ¿Cómo se les puede localizar?

Estos dos objetivos —recuerdo y aprendizaje— guían las teorías sobre la estructura de la memoria. Durante una conversación normal, determinadas zonas de la memoria se activan para entender las nuevas entradas. El que una experiencia recuerde a otras indica que la misma estructura se utiliza para procesar una y recordar la otra. Examinando el proceso de recuerdo exploramos al mismo tiempo la estructura de la memoria.



Armados con esta nueva concepción, volvamos al ejemplo del restaurante. Supongamos que al recibir la factura Pepe advierte un error y protesta. El camarero no sólo lo corrige, sino que además, para compensarle, le invita. Pepe —o cualquier sistema dinámico de memoria que intentase entender lo que ocurre— registrará en su memoria esta desviación de la rutina esperada. Si posteriomente le ocurre algo parecido mientras compra repuestos para su automóvil, recordará el episodio del restaurante



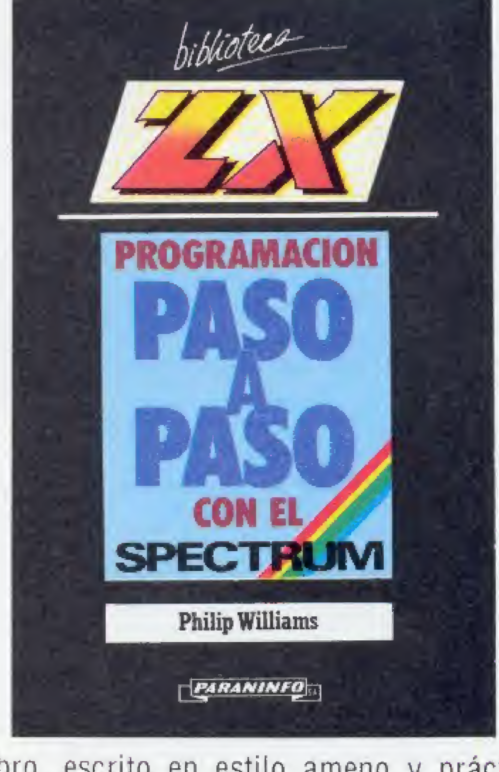
¡APROVECHA AL MAXIMO TU SPECTRUM!

Ahora, a tu alcance, dos obras fundamentales para que podáis sacar todo el partido posible a vuestro ordenador.



Esta publicación está diseñada para guiar al nuevo usuario del ZX Spectrum desde el momento que el ordenador se conecta hasta conseguir una base suficiente de la programación BASIC.
Incluye temas como:
- Introducción al teclado.
- Instrumentos útiles para la programación.
- Uso de comandos fáciles.
- Como construir un programa.
- Técnicas de programación.
- Aplicaciones prácticas.

100 pags. - 750 PTAS.



Este libro, escrito en estilo ameno y práctico, está dirigido a todos aquellos usuarios que han dejado atrás la etapa de los juegos y necesitan adentrarse en el fabuloso mundo de la programación.
El temario incluye:
- Reglas y herramientas del BASIC.
- La técnica de los organigramas.
- Cómo planificar un programa.
- El mundo de las rutinas.
- Variables y cadenas.
- Funciones matemáticas usuales.

109 pags. - 750 PTAS.

CUPON DE PEDIDO

Recorta este cupón debidamente cumplimentado y envíelo a INFODIS, S. A. C/ BRAVO MURILLO, 377-5.° A - 28020 MADRID

Sí, envíenme el(los) libro(s) que a continuación detallo al precio de 750 ptas. libro, más 100 ptas. en concepto de gastos de embalaje y envío.

El importe lo abonaré: POR CHEQUE ☐ CONTRAREEMBOLSO ☐ CON TARJETA DE CREDITO (VISA ☐) (AMERICAN EXPRESS ☐) (INTERBANK ☐)

Número de mi tarjeta

TITULO ______________________________

NOMBRE ______________________________

CALLE ______________________________

CIUDAD ______________ D. P. ______________

PROVINCIA ______________________________

Firma

OFERTA DE SUSCRIPCION

Todospectrum

REVISTA EXCLUSIVA PARA USUARIOS

Te ofrece la posibilidad de suscribirte con unas condiciones muy ventajosas para ti:

1 ***Recibir puntualmente, en tu domicilio la publicación TODOSPECTRUM que durante 12 meses te proporcionará lectura, programas, ayuda, entretenimiento, etc.***

2 *Consigue un práctico regalo:*



Una obra fundamental en la biblioteca de los aficionados al Spectrum:

- *Reglas y herramientas del Basic.*
- *La técnica de los organigramas.*
- *El fabuloso mundo de las rutinas.*
- *Variables y cadenas.*
- *Funciones matemáticas usuales.*

Esta publicación, escrita con estilo ameno y práctico, te ayudará a sacar todo el partido posible a tu máquina.

3 *La opción de ser protagonista. Tú puedes tener una participación directa con tus comentarios, programas, sugerencias, etc.*

4 *Obtener premios importantes con tus programas, y temas de interés.*

EN DEFINITIVA, TODO SON VENTAJAS

No dejes pasar esta oportunidad, suscríbete a "TODOSPECTRUM", cumplimentando hoy mismo el cupón de respuesta adjunto.



y esperará recibir una compensación. ¿Cómo puede recordarnos una experiencia sucedida en una tienda de repuestos otra ocurrida en un restaurante? Debe existir una estructura de memoria que interviene en ambos procesos.

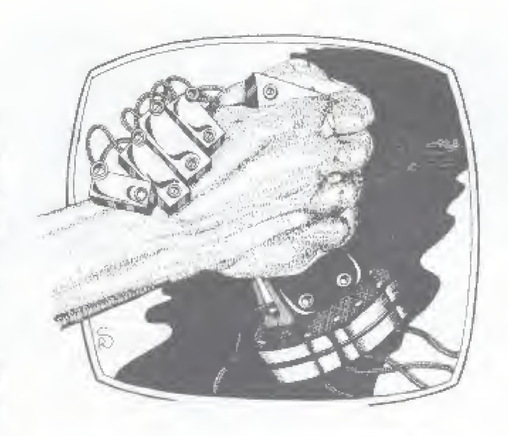
Indudablemente, no todas las espectativas son iguales en estos dos casos, pero algunas sí lo son. Los bloques están enlazados, es decir, existen sub-bloques de expectativas. En la terminología de característica de la inteligencia artifical los bloques se llaman MOPs (siglas de memory organization packages) y los sub-bloques escenas. Las escenas son compartidas, ya que aparecen en muchos MOPs.

En los programas y teorías sobre la inteligencia artificial se emplean más clases de memoria y estructuras de procesamiento de las que podríamos describir aquí. No obstante, los requisitos básicos para una organización útil de la memoria deberían haber quedado ya suficientemente claros.

Tomar en un momento determinado la porción necesaria de conocimientos entre los almacenados en nuestra enorme y cambiante memoria, es crucial para la comprensión del lenguaje. También lo es para mostrar el llamado sentido común. Aprender de las experiencias y aplicar los conocimientos en las situaciones pertinentes es el paso más importante hacia la inteligencia.

La inteligencia artificial

Se ha abusado tanto del término «inteligencia artificial» que es conveniente aclarar su significado. Los programas demostrativos de teorías sobre el conocimiento logran en ocasiones proezas impresionantes. Pero también lo consiguen otros programas que no pretenden ser modelos del conocimiento. La percepción pública de la inteligencia artificial se enfoca sobre todo



en la expresión «artificial». Los ordenadores hacen trabajos que nunca imaginamos que podría hacer una máquina. Su potencia y utilidad aumenta de día en día, pero esto no es inteligencia artificial.

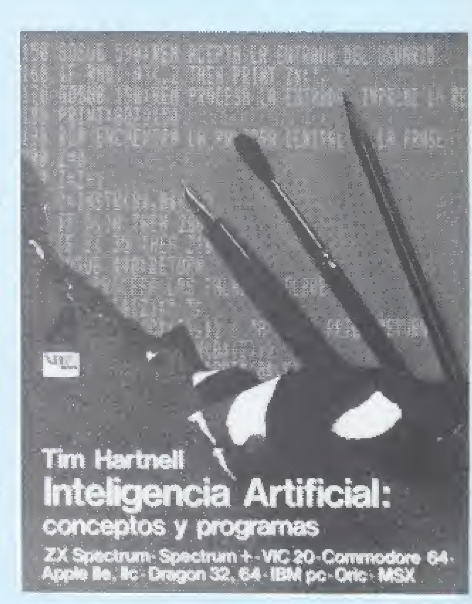
El verdadero sentido de la inteligencia artificial se centra realmente en la «inteligencia», algo que todavía es confuso y evasivo. Muchos programas «inteligentes» son prácticamente inútiles, mientras que otros muchos de gran utilidad y «listos» no son realmente inteligencia artificial. Si comprendemos esta diferencia nos ahorraremos numerosas confusiones.

Los sistemas expertos son un buen ejemplo. No razonan de la forma en que lo haría un experto humano. Aunque son programas que realizan trabajos útiles e importantes, en modo alguno sirven de modelo para las teorías sobre la inteligencia. Un experto humano no se limita a seguir una serie de reglas: tiene la experiencia suficiente para saber cuándo se enfrenta a un caso excepcional. Las teorías sobre la conducta experta deben describir el procedimiento que permite a un experto saber cuándo algo no se ajusta a las reglas y cómo obrar en esas circunstancia.

Aprender de la experiencia es el paso más importante hacia la inteligencia

Inteligencia artificial:

Tim Hartnell
Ed. Anaya Multimedia
267 págs., 1.300 ptas.



Quizá impresionados por lo ostentoso del término, la mayoría de los usuarios de ordenadores domésticos piensan que la Inteligencia Artificial queda lejos de sus posibilidades. Sin embargo, Tim Hartnell consigue en este libro demostrar todo lo contrario. Y lo que resulta más atractivo es que lo logra de un modo práctico, con programas «inteligentes» comentados detenidamente.

Los listados utilizan un BASIC estandarizado que se puede ejecu-

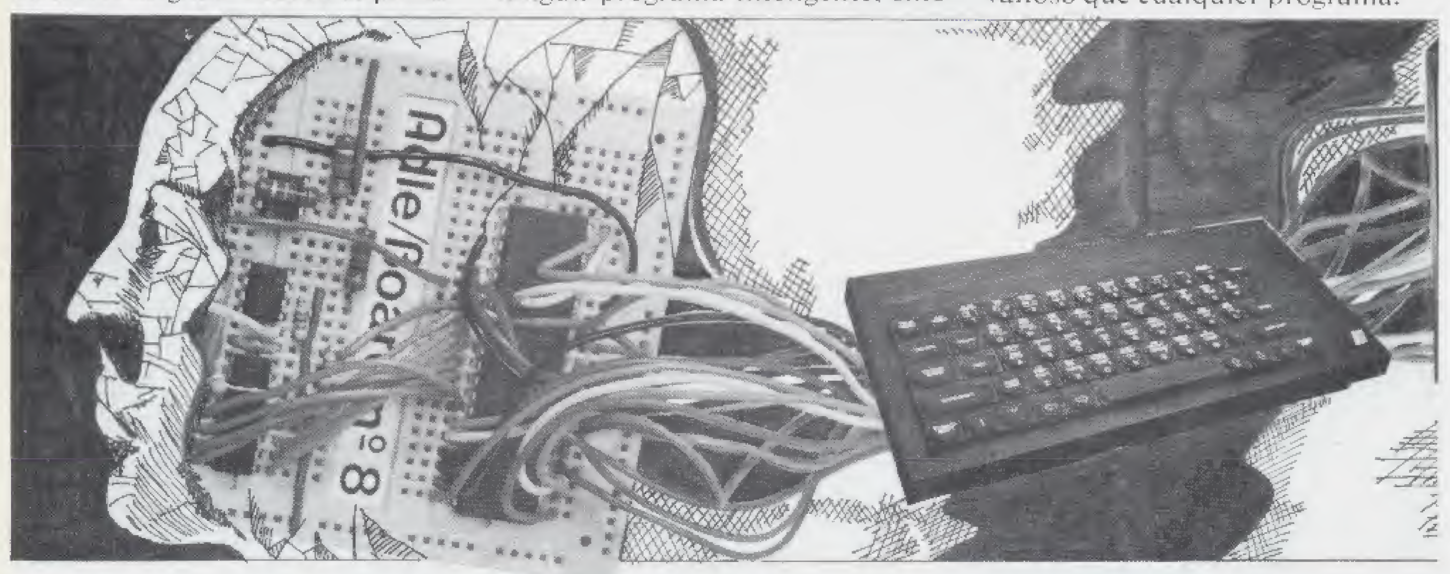
La inteligencia artificial es una ciencia mucho más básica de lo que generalmente se piensa. Muchas de las mejores ideas sobre el tema requieren un enorme esfuerzo para convertirse en aplicaciones útiles. Y en cierto modo, a partir de ese instante dejan de ser inteligencia artificial.

¿Tiene aplicaciones la inteligencia artificial? Sin duda. Avanzando en la investigación sobre el pensamiento se puede ayudar a la gente a pensar mejor. Probablemente la aplicación más importante no sea ningún programa inteligente, sino la posibilidad de averiguar cómo piensan los seres humanos.

El resultado de esta experimentación será una nueva forma de entendernos a nosotros mismos.

La inteligencia artificial forma parte del gran esfuerzo por comprender el pensamiento. Las investigaciones deben centrarse en la inteligencia, no en la máquina.

Los programas son sólo experimentos. El resultado será una nueva forma de entendernos a nosotros mismos. Y esto es mucho más valioso que cualquier programa.



conceptos y programas

tar en la mayoría de los microordenadores: Commodore 64, BBC, Vic 20, Dragon 32, Oric, TI 99/4A, TRS 80, Apple II, IBM PC y muchos otros. También, y digámoslo antes de que a alguien le dé un sofoco, en el Spectrum y Spectrum +. El apendice 4 contiene los listados con las modificaciones que necesita el Spectrum, que se reducen casi exclusivamente al tratamiento de cadenas.

El libro se divide en cuatro secciones dedicadas a distintas facetas del comportamiento inteligente: el pensamiento, la investigación, el habla y la ayuda. El lenguaje utilizado es sencillo y ameno, y la traducción, como ya es habitual en esta editorial, está bien realizada. Las primeras páginas de cada capítulo abordan por lo general aspectos teóricos, algunos de tanta importancia como los árboles de búsqueda, la comprensión del lenguaje natural y los sistemas expertos. Después se pasa al tratamiento práctico del tema, es decir, los programas y comentarios.

La calidad de los programas es más notable. Merece la pena destacar entre otros Tictac, juego de tres en raya que aprende a medida que juega; Bloquelandia, una sorprendente versión del famoso Shrdlu que se comunica con nosotros en castellano, y Doctor, versión en BASIC del no menos famoso Eliza, programa con el cual el ordenador se convierte en algo parecido a un psquiatra de la escuela rogeriana.

El libro termina con cuatro útiles apéndices. El primero da una serie de consejos para mejorar las técnicas de programación. El segundo se dedica a la información bibliográfica. El tercero es un pequeño vocabulario de términos informáticos y de inteligencia artificial. Y el cuarto y ultimo contiene las versiones para Spectrum de todos los programas que aparecen en el texto.

Un nuevo operativo para el Spectrum

Cuando una persona que ya sabe programar se encuentra por primera vez con un Spectrum, descubre que su BASIC es bastante potente, pero echa en falta algunas instrucciones muy populares en otros ordenadores. *Manuel Arana* ***nos presenta un listado que, una vez tecleado, permitirá a los usuarios de Spectrum 48K disfrutar de siete nuevos comandos directamente sobre el teclado. Con ellos esperamos contribuir a solucionaros muchos problemas.***

En este primer artículo dedicado a este programa me centraré únicamente en daros un método para cargarlo y en contaros detalladamente cómo funciona cada uno de estos comandos, cuál es su sintaxis, y qué posibilidades ofrece, dejando para números posteriores la aplicación de cómo funciona internamente, a nivel de lenguaje máquina. Esto nos dará pie a comprender un poco mejor el funcionamiento interno del Spectrum.

Antes de continuar, una advertencia. El programa tiene algunos problemas si lo queremos utilizar estando conectado el *Interface 1*, principalmente que se destruye cuando se comete algún error o intentamos usar el *interface*. De todas formas daremos unas pequeñas modificaciones que reducen estos problemas al mínimo, el programa únicamente se desconectará del resto del sistema cuando se produzca algún error del *interface* y en este caso será necesario volverlo a lanzar con una instrucción USR.

Cómo cargarlo

El proceso para cargar el programa es bastante sencillo, aunque un poco pesado. Lo primero que hay que hacer es teclear el programa del listado 1. Es «un poco» largo, pero no hay otra solución. El código máquina se ha incluido en forma de líneas DATA y se realizan controles de suma en cada línea para detectar los errores. Una vez tecleado todo el programa haz RUN. En la pantalla verás qué línea de datos se está cargando en ese momento, y te avisará en el caso de que exista algún error indicándote en qué línea está. Corrígelo y lanza otra vez el programa. Cuando ya no quede ningún error se detendrá después de «pokear» la última línea en el STOP de la línea 110. Ahora todo el programa está ya en la memoria y puedes hacer NEW. (No te preocupes, la situación está bajo control, pero si no te fías o piensas que se puede producir algún accidente guarda primero este programa en una cinta). Por último teclea el listado 2, que es un pequeño cargador, haz GOTO 100 y ya puedes grabar la copia definitiva del programa. Si piensas utilizar el *interface 1* utiliza el listado 3 en lugar del 2, y cuando se produzca un error del *interface* podrás volver a conectar con el programa mediante la instrucción RANDOMIZE USR 61260.



Mode

Cuando cargues el programa de la cinta te encontrarás con la sorpresa de que todo lo que tecleas aparece con unas letras mucho más estrechas de lo normal, a razón de 64 caracteres por línea. Este nuevo modo de presentación se puede controlar con la nueva instrucción MODE. La sintaxis de este comando es muy simple: MODE 0 coloca el Spectrum en el modo normal de presentación y MODE 1 selecciona el modo de 64 caracteres.

Trabajando en el modo de 64 columnas se pueden utilizar todas las posibilidades de la instrucción PRINT. Podemos hacer por ejemplo: PRINT AT 5,40;"*";TAB 50;"*". La única diferencia fundamental con el modo normal de presentación se refiere al uso de comas como separadores. En MODE 1 la pantalla está dividida en cuatro campos de 16 caracteres, y la utilización de la coma lo que



hace es colocar el cursor al principio del siguiente campo. De esta forma si escribimos una serie de datos separados por comas aparecen en la pantalla ordenados en cuatro columnas. También se pueden usar sin problemas los caracteres de control 8 y 9, que desplazan el cursor una posición hacia la izquierda y derecha respectivamente.

AUTO

Este comando se utiliza cuando se está tecleando un programa y su función es la de ir colocando los números de línea. Su sintaxis es: AUTO x,y donde x es el número de línea por el que debe empezar a numerar e y es el incremento entre dos números de línea consecutivos. Por ejemplo si hacemos AUTO 100,10 inmediatamente aparecerá en la parte inferior de la pantalla el número 100 seguido del cursor preparado para que empecemos a teclear la línea. Cuando pulsemos ENTER la línea será insertada en el programa y aparecerá el número 110 para que tecleemos la línea siguiente. Cuando terminemos, la forma de abandonar el modo AUTO es borrar el número de línea y pulsar ENTER.

El Spectrum provisto de «Arana Soft» dispone de dos modos de presentación: 32 ó 64 caracteres.

RENUM

Después de escribir y corregir un programa BASIC, cuando por fin funciona, normalmente nos encontramos con que en los números de línea reina la anarquía como resultado de haber borrado e insertado gran número de líneas. El comando RENUM hará que vuelva el orden. RENUM n,m renumera todo el programa comenzando por la línea n y con incrementos de m en m. También se encarga de cambiar los saltos que hubiera en el programa para que éste siga funcionando. Veamos un ejemplo. El siguiente programa calcula el factorial de un número de manera recursiva (procedimiento no muy recomendable en BASIC).

```
15 INPUT a:GO SUB 1000:PRINT a,f:STOP
1000 IF a<=1 THEN LET f=1:RETURN
1005 LET a=a-1
1007 GO SUB 1000
1020 LET a=a+1:LET f=f*a:RETURN
9000 SAVE "fac"LINE 15
```

Si queremos renumerarlo bastará con hacer RENUM 10,10 y

```
10 INPUT a:GO SUB 20:PRINT a,f:STOP
20 IF a<=1 THEN LET f=1:RETURN
30 LET a=a-1
40 GO SUB 20
50 LET a=a+1:LET f=f*a:RETURN
60 SAVE "fac"LINE 10
```

Quizá los amantes de la programación estructurada prefieran renumerar únicamente la subrutina. Para ello basta con hacer RENUM 1000,10,1000,2000 donde los dos primeros parámetros tienen el mismo significado que antes, el tercero indica la primera línea que debe ser renumerada, y el último la última línea que debe renumerarse. El programa quedaría

```
15 INPUT a:GO SUB 1000:PRINT a,f:STOP
1000 IF a<=1 THEN LET f=1:RETURN
1010 LET a=a-1
1020 GO SUB 1000
1030 LET a=a+1:LET f=f*a:RETURN
9000 SAVE "fac"LINE 15
```





Los comandos del BASIC que pueden ser modificados por una instrucción RENUM son todos los que pueden o deben llevar como parámetro un número de línea, esto es: GO TO, GO SUB, SAVE, LIST, LLIST, RESTORE, RUN y ON.

ELSE

La instrucción ELSE viene a completar la conocida construcción IF condición THEN comandos, que ahora puede tomar la forma IF condición THEN comandos ELSE otros comandos, que podríamos traducir como: si se cumple ésto haz ésto, si no haz ésto otro. De esta forma ELSE marca el punto donde debe continuar la ejecución en el caso de que no se cumpla la condición. Así podemos escribir la subrutina del ejemplo anterior en una sola línea:

```
1000 IF a<=1 THEN LET f=1:RETURN ELSE LET a=a-1:GOSUB 1000:LET a=a+1:LET f=f*a:RETURN
```

Cuando en una misma línea existan varias sentencias IF y varios ELSE puede surgir la duda de cuál de los IF se refiere a cada ELSE. La respuesta es que cada ELSE se refiere al IF más cercano a no ser que existan en medio otras instrucciones ELSE. De esta manera las construcciones IF-THEN-ELSE quedan anidadas unas dentro de otras.

El límite teórico de este tipo de estructuras anidadas es de 255, pero este límite es inalcanzable por la limitación de 127 sentencias en cada línea.

Por la forma en que está escrita la instrucción ELSE es posible introducirla en una línea en la que no aparezca el IF, pero en este caso no tiene ninguna utilidad ya que se comporta como si fuera una sentencia REM.

ON

Se trata de una nueva forma de saltos condicionados. La forma de utilizarlo es:

ON (expresión) GO TO línea1,línea2,...

El salto se producirá dependiendo del valor de la expresión, si vale uno saltará a la línea especificada



por el primer parámetro detrás del GOTO, si vale 2 al segundo y así sucesivamente. En el caso de que el valor de la expresión sea mayor que la cantidad de líneas indicadas después del GOTO el programa continuará por la siguiente línea. Este tipo de saltos también puede ser realizado con instrucciones del tipo GOTO calculado, pero este sistema presenta la ventaja de que es más comprensible y puede ser renumerado sin problemas. La instrucción ON se puede utilizar tanto con GOTO como con GOSUB.

Cuando tengas que teclear el GOTO o el GOSUB de una línea ON te encontrarás con el pequeño problema de que el cursor no está en K. Para solucionar esto las dos palabras han sido reasignadas a las mismas teclas pero en modo extendido y pulsando a la vez CAPS SHIFT. Sin embargo los que tengan el *Interface 1* no tendrán más remedio que hacer un poco de ejercicio con los dedos y poner dos puntos, GOTO o GOSUB y retroceder para borrar los dos puntos.



SWAP

Este comando sirve para realizar una transferencia cruzada sin necesidad de utilizar una variable intermedia. Para utilizarlo basta con hacer SWAP a,b y el resultado es el equivalente a hacer LET c=a-:LET a=b:LET b=c pero con la ventaja de que no utiliza la variable intermedia c y que es algo más rápido. Se puede utilizar tanto con variables numéricas como con alfanuméricas.



ON ERR

Esta instrucción le da al Spectrum una capacidad de tratamiento de errores de la que carecía to-

Con el comando ON ERR se puede evitar que el programa pare cada vez que encuentre un error.

talmente. La forma de utilizarlo es muy sencilla, basta con poner en el programa el comando ON ERR seguido de los códigos de error a

que nos queramos referir separados por comas. Esto marcará el punto del programa donde debe continuar la ejecución en el caso de que se produzca alguno de los errores indicados. No hay limitación en cuanto al número de errores en una instrucción ON ERR, ni en el número de veces que aparezca la sentencia ON ERR en el programa. El código del error producido queda almacenado en una variable llamada "err". La lista de los códigos de error la puedes encontrar en el libro de instrucciones del Spectrum en la página 189 y siguientes. La instrucción CONTINUE puede ser utilizada para que la ejecución vuelva al punto en que se produjo el error, de esta forma podrás hacer: ON ERR 1 CONTINUE, con lo que conseguirás que el programa no se pare porque alguien intente hacer BREAK.

Hay que hacer una advertencia en cuanto al uso simultáneo del error d (se produce al contestar no a la pregunta scroll?) y el error 1 (BREAK). Si ponemos ON ERR d,1 CONTINUE y cuando se produce el mensaje *scroll?* respondemos pulsando simultáneamente CAPS SHIFT y SPACE se producirá un error d que será tratado como tal, pero en el nomento en que comienza a ejecutarse la instrucción CONTINUE se produce un error 1 (BREAK) y ese será el punto donde saltará el programa después de CONTINUE, quedando la máquina «colgada» en este bucle.

La instrucción ELSE mejora la estructuración de nuestros programas.



Para terminar incluyo un pequeño programa que muestra cómo se puede utilizar la instrucción ON ERR para solucionar los problemas que se presentan al representar gráficamente una función. Se produce el error B cuando un punto queda fuera de los límites en cuyo caso optamos por ponerlo en el borde de la pantalla. También se puede producir un error matemático por no estar definida la función en ese punto (error A) o por intentar dividir por cero (error 6). En cualquiera de estos dos casos ignoramos el punto.

TABLA 1

Sintaxis	Acceso teclado	Con Interface 1
MODE n	CS + Y	CS + 6
SWAP v1,v2	CS + U	CS + 5
ON n GOTO 11,12,...	CS + T	CS + 4
RENUM 11,n i,12,13¿	CS + R	CS + 3
AUTO 11,n	CS + E	CS + 2
ON ERR e1,e2,...	CS + W	CS + 1
ELSE	CS + Q	CS + 0

n - expresión numérica
11,12,13 - número de línea o expresión numérica
v1,v2 - nombres de variables numéricas o alfanuméricas
e1,e2 - códigos de error (una cifra o una letra)

Para los que todavía no lo hayan descubierto la tabla 1 es un resumen de la sintaxis de cada comando e indica la forma de acceso a las palabras clave. Recuerda que todas ellas se consiguen en modo extendido.

Manuel Arana



N.º 4
PRIMERA REVISTA EN CASSETTE
ABRIL-MAYO 85
SPECTRUM
COMPUTING
PARA 16K Y 48K
3 D
Añada una nueva dimensión a su SPECTRUM.
Acción en tres dimensiones.
Busque y destruya la flota de ataque estelar.
Piérdase en nuestro laberinto en tres dimensiones en código máquina.
Defensa
Su oportunidad para venir en defensa del planeta.
875 Ptas.
Más sobre la programación de dibujos animados.
Clocks up
¿Sabe generar un reloj digital en su SPECTRUM?
Y MUCHO MAS
BIENVENIDO A
SPECTRUM
COMPUTING
LABERINTO EN TRES DIMENSIONES
GRAFICOS
WIZARD
RELOJ
DEFENSA
DIBUJOS ANIMADOS
ATAQUE ESTELAR
AGENDA TELEFONICA
SKI
PUZLE
LA SERPIENTE
MAS DE 150.000 PTAS. EN PREMIOS. BASES EN EL INTERIOR
Solicítela a: INFODIS C/ Bravo Murillo, 377 - 5.º A - 28020 MADRID
CUPON DE PEDIDO
Nº 4
Sí, envíenme al precio de 875 ptas.... ejs. de SPECTRUM COMPUTING
El importe lo abonaré: Contra reembolso □ Con mi tarjeta de crédito □
Adjunto cheque □ American Express □ Visa □ Interbank □
Número de mi tarjeta
Fecha de caducidad
NOMBRE
DIRECCION
CIUDAD
PROVINCIA
Sin gastos de envío

LISTADO 1

```
  10 CLEAR 60158: LET n=60159: R
ESTORE
  20 PRINT AT 5,5;"Cargando line
a ";
  30 FOR i=1000 TO 2600 STEP 10
  40 PRINT AT 5,20;i
  50 READ a$,a: LET s=0
  60 FOR j=1 TO LEN a$-1 STEP 2
  70 LET d=16*(CODE a$(j)-48-39*
(a$(j)>"9"))+CODE a$(j+1)-48-39*
(a$(j+1)>"9")
  80 POKE n,d: LET n=n+1: LET s=
s+d: NEXT j
  90 IF s<>a THEN  PRINT AT 5,5;
"Error en la linea ";i: BEEP .5,
-10: STOP
 100 NEXT i
 110 BEEP 1,-10: STOP
1000 DATA "01ebf5e52a785c2322785
c7cb52003fd3440c5d5cd19ebc34d00c
d8e02c02100",3675
1010 DATA "5ccb7e200723352b20023
6ff7d21045cbd20eecd1e03d071005cb
eca1003eb21",2897
1020 DATA "045cbeca1003cb7e2004e
bcb7ec85f7723360523a095c7723fd4
e07fd5a01e5",3199
1030 DATA "cd65ebc306037bfe3adaa
4eb0dfa4f032803c64fc921eb0104280
d210502f578",3794
1040 DATA "fe1928032149ebf1c34a0
37edcda5c1a7becedbfaeeaabdddedf7
f18b7c1c1d",4185
1050 DATA "db19d91ed7fe30d80dfa9
d03c28903215402cb68ca4a03fe38d28
203e607320d",3639
1060 DATA "5c3e1004c83cc90000000
0000000000202020200020000050500
00000000000",955
1070 DATA "507050507050000020704
07010702000404010204010100020502
060b0f00000",1952
1080 DATA "2040000000000000020404
0404020000040202020204000000000502
07020500000",1008
1090 DATA "002020702020000000000
00020204000000000070000000000000
00060600000",672
1100 DATA "101020204040000070505
05050700000206020202070000020501
02040700000",1440
1110 DATA "601060101060000010705
05070100000704060101060000030406
05050200000",1488
1120 DATA "701020204040000070502
05050700000205050301060000000002
00000200000",1232
1130 DATA "002000000202040000107
04020100000000070000700000000000402
01020400000",752
1140 DATA "205010200020000060f0d
0a080700000205050705050000060506
05050600000",2128
1150 DATA "205040405020000060505
05050600000704060404070000070407
04040400000",1856
1160 DATA "205040705020000050507
05050500000702020202070000010101
05050200000",1504
1170 DATA "505060605050000040404
04040700000507070705050000070505
05050500000",2032
1180 DATA "205050505020000060505
06040400000705050507070100070505
06060500000",2000
1190 DATA "304020101060000070202
02020200000505050505070000050505
05050200000",1488
1200 DATA "507070707020000050502
02050500000505050202020000070102
02040700000",1648
1210 DATA "704040404070000000406
02030100000701010101070000020702
02020200000",1296
1220 DATA "00000000000f00020504
0f040f00000060107050700000040406
05050600000",1856
1230 DATA "003040404030000010103
05050300000002050604030000030406
04040400000",1296
1240 DATA "003050503010600040406
05050500000200060202070000010001
01010502000",1312
1250 DATA "405060605050000040404
04040300000050707070500000000605
05050500000",1776
1260 DATA "002050505020000000605
05060404000003050503010100000304
04040400000",1376
1270 DATA "003040201060000020702
02020100000050505050700000000505
05050200000",1296
1280 DATA "005070707020000000505
02050500000050505030106000000701
02040700000",1536
1290 DATA "302040202030000020202
02020200000602010202060000050a00
0000000060",1088
1300 DATA "80b0a0b0809060f090909
0909090f030303030000000000c0c0c0c
000000000f0",3552
1310 DATA "f0f0f0000000000000000000
0303030303030303030303030c0c0c0c
030303030f0",2496
1320 DATA "f0f0f03030303000000000
0c0c0c0c0303030300c0c0c0c0c0c0c0c
0c0c0c0c0f0",4416
1330 DATA "f0f0f0c0c0c0c00000000
0f0f0f0f030303030f0f0f0f0c0c0c0c
0f0f0f0f0f0",5568
1340 DATA "f0f0f0f0f0f03eeaed4
7ed5eed7b3d5cf1c335fcaf32f8f0fd3
63102cdf0f1",5626
1350 DATA "cdb0163af8f0a728193ef
fcd01162af1f0ed5bf3f019cb7420082
2f1f0444dcd",4309
1360 DATA "1b1a3e00cd0116cd73f3c
d89f4fdcb007e2012fdcb306628772a5
95ccda711fd",3754
1370 DATA "3600ff18dd2a595c225d5
ccd75f478b1c2f9f0dffe0d289dfdcb3
046c4b7f1cd",4628
1380 DATA "6e0dcddaf13e19fd964ff
d7752fdcb01fefd3600fffd360a01e57
a8a5c22b05c",4204
1390 DATA "e1cd52f53a3a5c3c28264
fc52a535c1e19cd51f1ebe3eb3816c5c
d67f4cd90f9",4300
1400 DATA "c1caacf0ebe3ebcd68f1e
be3eb30eac176fdcb01aefdcb304ec4d
c0efd7e003c",5426
1410 DATA "f5210000fd7437fd74262
20b5c21010072165ccdb016fdcb37aec
d6e0dcddaf1",3508
1420 DATA "fdcb02eef147fe0a3802c
607cdef153e20d778119113cd0a0caf1
13615cd0a0c",3331
1430 DATA "ed4b455ccd1b1a3e3ad7f
d4e0d0a00cd1b1acd9710fd7e003cc48
6f0fd3e0aff",3627
1440 DATA "fdcb019ec381effe09280
4fe15200Ifd340dC10300117O5c21445
cfdcb0e7e28",3163
1450 DATA "0109edb8c97e10010000c
31ff02a5d5ce5d5cdb01a31edf0225d5
ccd1f1cd17b",3574
1460 DATA "fe15200bafdbfee61ffe1
f20f73e15cd86f0cd282dcdff2ae1225
d5cd13e0192",4107
1470 DATA "32445c21e3efe5fd3600f
fc33ff54552522a00000a004552520 0e
d43495c2a5d",3125
1480 DATA "5ceb21e4f0e52a615c37e
d52e56069cd6e192006cdb819cde819c
1793db02878",3913
1490 DATA "c503030303Zbed5b535cd
5cd551ee122535cc1c5132a615c2b2be
db82a495ceb",3303
1500 DATA "c1702b712b732b72f1c35
8ef7efe3a2818237ee6c037c046234ee
d43425c234e",3629
1510 DATA "2746e509444de11600c5c
d72f1c1d0606918de225d5c0e0015c8c
d67f4bb2004",3569
1520 DATA "a7c9237ecdb618225d5cf
e2220010dfe3a281afecb2816fe19201
6cb412012cd",3246
1530 DATA "67f4d5cd90f9d12a5d5c2
b18c6cb4128c7fe0d20cd1537c921000
0227d5cfdcb",3887
1540 DATA "308ecddaf13efecd0116c
d4d0d0618cd440e2a515c11cdfcc3cf0
d3efdcd0116",3649
1550 DATA "2a515c11cdfca77323722
31109f3c3a70ded733f5cfd360210cdb
7f1c3a017fd",3891
1560 DATA "cb01c6d5ebfdcb3096213
b5ccb96fdcb376e2802cbd62a5f5ca7e
d5220053e3f",3998
1570 DATA "cdeff2cdb6f2eb7ecdb61
823fe0d2806ebcd37f218e0d1c9cd882
c3fd26c19fe",4875
1580 DATA "2028f9fe2c28f5fe1ac33
e19af3210fdcd4d0dfdcb029efdcb02a
e2a8a5ce52a",4046
1590 DATA "3d5ce5219ff2e5ed733d5
c2a825ce537cd9511ebcdfef1ebcdb6f
22a8a5ce3eb",5002
1600 DATA "cd4d0d3a8b5c923811200
67bfd965030093e20d5cdcdfcd118e93
e20d7c37c11",3584
1610 DATA "1600fd5efe21901acdb50
3fd3600ffed5b8a5c1cc37e112a5b5ca
7ed52c03a41",3727
1620 DATA "5ccb072804c6431821213
b5ccb9e1b1a13fe2c2802fe193e4b280
4cb56280bcb",2628
1630 DATA "de3cfdcb305e28023e43d
5cdeff2d1c9d921915c56d53604cdcdf
ce1fd7457d9",4764
1640 DATA "c9cdb7f1cd6e0dc3daf1f
dcb025ec44bf2a7fdcb016ec83a085cf
dcb01aef5fd",5098
1650 DATA "cb026ec403f3f1fe18304
7fe103030fe06300a47e6014f781fc61
2181f200921",2956
1660 DATA "6a5c3e08ae77180efe0ed
8d60d21415cbe7720023600fdcb02deb
fc9fd71d311",3563
1670 DATA "63f318063a0d5c1109f32
a4f5c232373237237c92a3d5ce52146f
4e5ed733d5c",3112
1680 DATA "cdd415f51600fd5eff21c
800cdb503f1217ff3e5fe0a285efe0b2
83dfe082814",3888
1690 DATA "fe182816fe092821fe0cc
24c0fcd04f4ebc3e519cd04f4c311103
e3acd810f3e",3576
```

```
1700 DATA "08cd810f3e18c3810f2a5
b5c7efe0dc8545d23cd32f4c31110fdc
b376ec02a49",3467
1710 DATA "5ccd6e19ebcd9516214a5
ccd1c19cdf0f13e00c30116fdcb376ec
2011021495c",3496
1720 DATA "cd0f1918e92a5b5c37cd9
511ed521923c1d8c5444d626b231acd3
2f4e6f8fe10",3796
1730 DATA "2009231ad617ce0020012
3a7ed4209eb38e3c9fe3ac07efe08200
a237efe1820",3216
1740 DATA "032318012b1ac9fdcb306
6ca2610fd3600ff1600fd5efe21901ac
db503c377f3",3529
1750 DATA "2a5d5c7ecd6cf4d0cd740
018f7fe203fc8fe18c37f002a595c2b2
25d5ccd67f4",3896
1760 DATA "c3031adfb9c28a1c18def
dcb01becd75f4af32475c3d323a5c180
3cd67f4cdbf",4075
1770 DATA "16fd340dfa8a1ccd5ff40
600fe0dca7bf5fe3a28e6213ff5e54fc
d67f479fece",4507
1780 DATA "3011d6183804fe083804c
38a1c7e21f1f5180f21ccf54fcddc163
8f17921481a",3287
1790 DATA "d6ce4f094e0918032a745
c7e2322745c01e7f4c54ffe20d282f4f
e03200ddfcd",3627
1800 DATA "de1cbfc1ccb6f5ebc3161
ca728f4fe0520030518eefe022008c1c
d561ccdb6f5",4197
1810 DATA "c9fe07c2631bfdcb017ef
dcb0286c44d0df13a745cd613cdfc21c
db6f5c3ad1c",4501
1820 DATA "cd541fd27b1bfdcb0a7e2
0712a425ccb7c281421feff22455c2a6
15c2bed5b59",3432
1830 DATA "5c1b3a445c1833cd6e193
a445c2819a72043477ee6c078280fcff
fc1cd3025c8",3240
1840 DATA "2a555c3ec0a6c0affe01c
e0056235eed53455c235e2356eb19232
2555ceb225d",3281
1850 DATA "5c571e00fd360aff15fd7
20dca9af414cd72f12808cf16cd3025c
0c1c1dffe0d",3997
1860 DATA "28bafe3aca9af479a728f
9cf0bf708fb09fd0aea0bfa0ce60deb0
ef50fee10e9",4457
1870 DATA "11f012e313e414fc15e01
6e117d818ce1900616d575a6062646a1
21416181a1e",2930
1880 DATA "2027292b2d2f313336383
a3d0365f60000f30370f6057af506cb0
58df60035fc",2808
1890 DATA "043d06cc0605f2f6050bf
7051df7054df80567f805cef80596f80
905a8f80507",3308
1900 DATA "f70563f80905bdf805e0f
8080040f9080586fa05bcf9053afa05c
4f9060062f9",3803
1910 DATA "0581f900cbfccd671e010
000cd451e1803cd991e78b12004ed4bb
25cc5ed5b4b",3411
1920 DATA "5c2a595c2bcde519cd00f
3c3c61ec1cd3025280aef0238ebcde93
4daa3f6cd5f",4170
1930 DATA "f4c39df4fd460d0e01cd5
ff4fe1820030d2811fefa20010c04cdc
cf6cd740030",3695
1940 DATA "e8c37bf5fd700dcd67f4c
39df47ecde3f6cdb618225d5cfe3ac8f
ecbc8fe0d37",5246
1950 DATA "c82318e9fe22c0237efe2
2c8fe0d20f7f137c9dffecd200ae7cd8
21ccdb6f5c3",4809
1960 DATA "161dcdb6f5c3131d3e031
8023e02cd3025c40116cd4d0dcddf1fc
db6f5c9cd30",3430
1970 DATA "2528083e01cd0116cd03f
3fd360201cd55f7cdb6f5ed4b885c3a6
b5cb838030e",3360
1980 DATA "2147ed43885c3e1990328
c5cfdcb0286cdd90dc303f3cd4e2028f
bfe28200ee7",3794
1990 DATA "cddf1fdffe29c28a1ce7c
346f8feca2011e7cd1f1cfdcb37fefdc
b0176c28a1c",4781
2000 DATA "180dcd8d2cd243f8cd1f1
cfdcb37becd3025ca46f8cdbf1621715
ccbb6cbee01",4210
2010 DATA "0100cb7e200bfd7e01e64
020020e03b677f7360d790f0f30053e2
2122b77225b",2318
2020 DATA "5cfdcb377e202c2a5d5ce
52a3d5ce521cef7e5fdcb30662804ed7
33d5c2a615c",3786
2030 DATA "cda711fd3600ffcd73f3f
dcb01becdb9211803cd73f3fd362200c
dd621200acd",4214
2040 DATA "4bf2ed4b825ccdd90d217
15ccbaecb7ecbbe201ce1e1223d5ce12
25f5cfdcb01",4223
2050 DATA "fecdb9212a5f5cfd36260
0225d5c18172a635ced5b615c37ed524
44dcdb22acd",3416
2060 DATA "ff2a1803cdfc1fcd4e20c
a55f7c9cdb228c28a1ccd30252008cbb
1cd9629cdb6",4138
2070 DATA "f5c3152c3e0318023e02f
d360200cd3025c40116dfcd70203814d
ffe3b2804fe",2960
2080 DATA "2c2006e7cd821c1808cde
61c1803cdde1ccdb6f5c32518dfcd302
5c2371ecdfb",3789
2090 DATA "24fe2cc4b6f5e718f5dff
e2c2806cdb6f5c37724e7cd821ccdb6f
5c39423dffe",5087
2100 DATA "2cc28a1ce7cd821ccdb6f
5c32d23e7cd1f1ccd3025c2f51ddffe2
c28f1cdb6f5",4592
2110 DATA "c9dfcd3025c2651ffdcb0
1f6cd8d2c3016e7fe242005fdcb01b6e
7fe28203ce7",4248
2120 DATA "fe292820cd8d2cd28a1ce
be7fe242002ebe7eb010600cd5516232
3360efe2c20",3411
2130 DATA "03e718e0fe292013e7fe3
d200ee73a3b5cf5cdfb24f1fdae01e64
0c28a1ccdb6",4307
2140 DATA "f5cd991ec5cd991ee1505
9eba7ed5222f1f0ed53f3f021f8f0360
1ed7b3d5cf1",4981
2150 DATA "c35fefcdd52d3804fe023
802cf0a874f0600217df9095e2356ed5
30cfdc9f409",3473
2160 DATA "35fdcd3025c27af5cd90f
9cd5ff4c39df4cd5ff4cd882c30d3f50
8f1fe3a3804",4949
2170 DATA "e6dfd607d630bb200108c
d67f4fe2c200acd67f4cd882c38e3cf0
b08c9cd3025",3993
2180 DATA "c27af518c6cd30252019c
d1f1c0e2ccd82f43a3b5cf5cd1f1cf1f
d5601aae640",3794
2190 DATA "c8cf0b2a5d5ce5cdfb24f
dcb0176cc0afacd67f4cdfb24e1225d5
ccd1f1ccdff",4617
2200 DATA "2acd67f4cd1f1ccdff2ac
9cdf12b78b1cab22ae1d5f7d1c52a615
cebedb0cd67",5047
2210 DATA "f4cdfb24c1e1c5225d5cc
d1f1ccdff2aed5b615cc1cdb22ac300f
acd5ff4cd82",4795
2220 DATA "1c47e6fefeec20990e00c
5cd302537c4d52d30023efff5cd67f4c
d821cf1cd30",4289
2230 DATA "2528083d281cf5ef0238f
147cd5ff4fe2c7828e2f1cd3025c8d0f
eedcaed1ec3",4390
2240 DATA "671ec10dc518e4cd30252
008dffe2cc0e7c37a1cdffe2c2809211
027e5010000",3295
2250 DATA "180be7cd7a1ccd991ec5c
d991ee1a7ed4209380e222dfced432ff
ccd991e78b1",3987
2260 DATA "2002cf0aed4333fccd991
eed4331fced4b2dfccd02fcdde5ccb81
956235ee1a7",4389
2270 DATA "ed5230de2a5d5c225f5cc
d2cfb3efecd01162a5f5c225d5cfd362
6002a2dfc23",3248
2280 DATA "cd6e197ef53e80772a2ff
ccd6e19ed5b31fcd57efe802811d1722
3732be52a33",3946
2290 DATA "fc19e3cdb819eb18ead1f
177c92a535c7ee6c0c023234e2346e50
9224d5ce122",4091
2300 DATA "2bfc225d5c180ee11602c
d72f130062a4d5c2318dbcd67f4e5216
afb4fcddc16",3601
2310 DATA "30e516005e19cd2c1618d
cf8101c25ec3aed38e136f034f732e53
00001caff2a",3494
2320 DATA "5d5c237ecdb618fe0dc8a
0b920f4225d5c181701ecfecd7efbe7c
d821cf5ef02",4264
2330 DATA "38f1fe2cc0cda9fb18ef2
a5d5c237efe0e2806cd1b2dd818f4e5c
db618fe3a28",4130
2340 DATA "09fe0d2805fe2cd1c0d52
b2b462b4ecd02fce1d8dde5c1e523232
371237e1ed",4109
2350 DATA "5b5d5c13225b5cc5cde51
93effcd0116c1cd1b1a2a4d5ced5b2bf
ca7ed52eb72",3833
2360 DATA "2b73c92a2dfca7ed42d82
a2ffccd1cff0009dd2a31fcc5cd6e19c
1cd8019d0ed",4319
2370 DATA "5b33fcdd19c5cdb819eb1
8ee000010270000a000a002ab25c363
e2bf92b2b22",2663
2380 DATA "3d5cfd213a5c21b65c224
f5c11af15011500ebedb0eb2b22575c2
322535c224b",2828
2390 DATA "5c36802322595c360d233
6802322615c22635c22655c3e0f328d5
c328f5c3248",2285
2400 DATA "5c3e01d3fefd35c6fd35c
acd12ff11105c010e00edb0fdcb01cec
ddf0efd3631",4124
2410 DATA "02cd00f3af11b5fccd0a0
cfd360100fdcb02eec35fef807f20313
9383420417 2",3547
2420 DATA "616e6120534f46545741 5
2c5cf0bcdf0fce5f57e32a5fef1fe203
025fe103821",4038
2430 DATA "fe18380ad61811ebfecd1
1fde1c9e11845210efdfdcb014ec023f
dcb0246c823",4132
2440 DATA "c92a0cfdcd2c16e13aa5f
e77c935fd000000f5cd410c3e20fdcb0
146cc3b0c1a",3556
2450 DATA "e67fcd3b0c1a138730f5d
1fe482803fe823e20c33b0ccd030bfe2
0d21bfefe10",3694
2460 DATA "386afe183062fe1630531
14ffdc37d0a11cdfccd800a2a0e5c573
a0dfdfe09ca",3513
```

```
2470 DATA "910a7dfe16da1122201e4
44a3e3f91daac0ac6044fafcb3938023
e1532a5fecd",3230
2480 DATA "a30acdf0fcc307fd7ccd0
30bf53aa5fefe01cb11f181d603e63fc
3ca0a11a1fd",4583
2490 DATA "18ab114ffdc3700a3e3f1
870fe06200bcb213e4491e630c61018c
efe08202e3a",3056
2500 DATA "a5fea72805af32a5fec93
e1532a5fe0c3e22b92015fdcb014e200
9040e023e19",3057
2510 DATA "b82007050e21af32a5fec
3d90dfe092013fd7e57f5fd77573e20c
d41fef1fd77",4059
2520 DATA "57c3dc0afe0d20a0fdcb0
14ec20affaf32a5fe0e21cd550c05c3d
90dcd21fec3",4075
2530 DATA "dc0afe90381cd6a5d25f0
bc615f53aa5fea728053e20cd41fef1c
5ed4b7b5cc3",4338
2540 DATA "6a0bc501c6eaeb213b5cc
b86fe202002cbc626006f29292909c1e
b3aa5fea728",3617
2550 DATA "06af32a5fe181f793d3e2
1200e054ffdcb014e2806d5cd0affd17
9b9d5cc550c",3405
2560 DATA "d13e1532a5fec5e5fd7e5
706ff1f3801041f1f9f4f3e08a7fdcb0
14e2805fdcb",3579
2570 DATA "30ce37eb0818003e0fa14
f3ef0b047afed671aa0aea912afed6f1
80d3ef0a14f",3606
2580 DATA "3e0fb0471aa0aea912083
81814233d20d3eb25fdcb014eccdb0be
1c13aa5fea7",3621
2590 DATA "c00d23c9083e20835f081
8e1bf454c53c54f4e204552d2415554c
f52454e55cd",3152
2600 DATA "4fce535741d04d4f44c5c
021a5fe3600c3cd0e21cdfc22c55c21c
615c9ed42c83fd1d8d5c9",4932
```

LISTADO 2

```
10 PAPER 1: BORDER 1: INPUT ;:
INK 9: CLEAR 60158: PRINT AT 5,
10;"EXT. BASIC";AT 7,12;"espera"
20 PLOT 72,144: DRAW 96,0: DRA
W 0,-40: DRAW -96,0: DRAW 0,40
30 LOAD ""CODE
40 RANDOMIZE USR 61254
100 SAVE "EXT" LINE 10: SAVE "E
XT"CODE 60159,5157
```

LISTADO 3

```
10 PAPER 1: BORDER 1: INPUT ;:
INK 9: CLEAR 60158: PRINT AT 5,
10;"EXT. BASIC";AT 7,12;"espera"
20 PLOT 72,144: DRAW 96,0: DRA
W 0,-40: DRAW -96,0: DRAW 0,40
30 LOAD *"m";1;"EXT."CODE
40 POKE 23807,205: POKE 23808,
252
45 POKE 61265,24: POKE 61266,1
2
50 RANDOMIZE USR 61260
100 SAVE *"m";1;"EXT" LINE 10:
SAVE *"m";1;"EXT."CODE 60159,515
7
```

LISTADO 4

```
10 DEF FN x(x)=(x-xmin)/(xmax-xmin)*256
20 DEF FN y(y)=(y-ymin)/(ymax-ymin)*176
30 CLS : INPUT "Funcion : "; LINE a$
40 INPUT AT 3,0;"Y max :";AT 0,0;"X min :""X max :""Y min :"
;AT 0,8;xmin;AT 1,8;xmax;AT 2,8;ymin;AT 3,8;ymax
50 LET a=FN x(0): IF a>=0 AND a<=255 THEN PLOT a,0: DRAW 0,175
60 LET a=FN y(0): IF a>=0 AND a<=175 THEN PLOT 0,a: DRAW 255,0
70 FOR x=xmin TO xmax STEP (xmax-xmin)/256
80 LET x1=FN x(x): LET y1=FN y(VAL a$): PLOT x1,y1
90 ON ERR 6,a
100 NEXT x
110 STOP
120 ON ERR b
130 IF x1>255 THEN LET x1=255
140 IF x1<0 THEN LET x1=0
150 IF y1>175 THEN LET y1=175
160 IF y1<0 THEN LET y1=0
170 CONTINUE
```



Suplemento especial Septiembre 1985

ARCHIVE

Base de datos inteligente

Análisis de software

CARTRIDGE DOCTOR

Programas de gestión

CONTROL DE STOCKS Y FACTURACION



El QL Español, sin más retrasos

Ya hemos podido ver que el QL en su primera versión española, que responde a VER$="MGE", probablemente en honor de alguno de los que participaron en su conversión a las lenguas peninsulares. Con un teclado que permite la introducción de los caracteres propios sin necesidad de malabarismos con el teclado, y una nueva versión de los programas de **Psion,** íntegramente convertidos al castellano y mucho más rápidos, el QL queda en disposición de luchar con máquinas mucho mayores (hay que tener en cuenta que el sistema **Xchange,** nombre de los programas de **Psion** en versión IBM PC, cuesta más que el QL, que los incluye con la máquina).

Por otro lado, finalmente Sinclair ha optado por adoptar como oficial el sistema de discos de 3 1/2 pulgadas de **Micro-Peripherals,** con unidad simple o doble de 800 K formateadas.

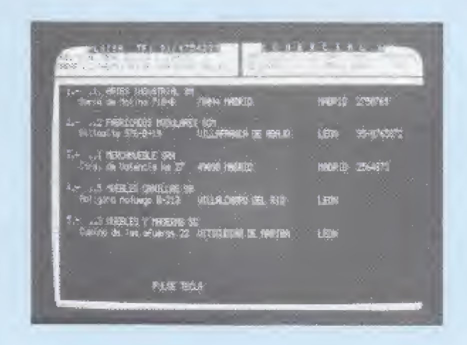
Problemas con la impresora

Algunos usuarios nos han comunicado que han tenido problemas para sacar por impresora los caracteres españoles. Aunque el manual no lo mencione, existe una instrucción (TRA 1) que hace que el programa convierta su representación interna para estos caracteres en el standard ASCII. Así, cualquier impresora que tenga seleccionado el juego de caracteres español los imprime sin problemas, tomando la precaución de teclear TRA 1 en el BASIC antes de imprimir.

Además, la nueva versión incluye una serie de procedimientos y funciones que nos ayudan a controlar los errores en SuperBASIC, y que comentaremos en un próximo número.

Programas españoles para el nuevo QL

Existen al menos dos compañías que se han tomado muy en serio las posibilidades profesionales de esta máquina: se trata de Alsi Comercial, y de Takis. **La primera ya nos anunció hace tiempo la conversión de su programa de gestión integrada para Spectrum.**



Recientemente hemos podido ver una versión demo **de este programa, y su lanzamiento definitivo se hará muy en breve. Además, se nos anuncian otros programas:** Alsistocks **destinado a control de almacén, con hasta 1.800 artículos en** microdrive **o 14.000 en disco. Esperamos verlo dentro de poco.**

CambiAlsi **se encarga de gestionar nuestras facturas o letras de cambio, y** AlsiMail **sirve para manejar un fichero de direcciones e imprimir etiquetas o recibos. Con** Alsicont **podremos llevar la contabilidad, con opciones interesantes. Finalmente, su** Simulador Financiero **nos ayuda a tomar decisiones y planificar nuestro negocio.**



Takis, **que ya había realizado algunas aplicaciones profesionales para Spectrum, también confía en la nueva máquina. Hemos podido ver su** MASTER-COM, **un programa que combina la contabilidad con el control de** stocks.

El programa ofrece una presentación muy atractiva, e intentaremos hacerle un test **exhaustivo en breve. Está preparado para su uso con la unidad de** diskettes **que comercializa** Investrónica, **y dispone de hasta 10.000 artículos, 1.000 proveedores y 5.000 clientes o distribuidores. La impresión se podrá realizar por cinco modelos de impresora diferentes, y es el primero de una serie de programas profesionales. Su siguiente «incursión» será en el terreno de las profesiones liberales, con programas para médicos, abogados, etcétera.**



Todos sus programas trabajan con la unidad de disco, y dispondrán de protección Hardware, **ya que, nos han confesado, están muy escamados con la piratería de los programas para Spectrum.**

También en Inglaterra hay movimiento

Entre las compañías de *software* que han elegido el QL, **Talent Software** ha producido ya dos programas muy interesantes: **Cartridge Doctor,** destinado a «curar» nuestros cartuchos de *microdrive* que presenten problemas de carga, y del que hablamos en otro artículo en este mismo número. Por otro lado, **GraphiQL** es un programa de dibujo que nos permite dibujar con el ordenador, como si se tratara de un lienzo. Novedades respecto a otros programas del mismo tipo son la definición de texturas (análogo del dibujo de un papel pintado), con las que se pueden rellenar áreas, y el dibujo con aerógrafo, que rellena la zona por la que nos movemos.

Ambos juegos son distribuidos por **Serma,** que también comercializa la versión para QL de **D-Day,** un juego de guerra de los más potentes que se han desarrollado, que nos permite dirigir el ejército aliado y alemán en el desembarco de Normandía.



APLICACION

ARCHIVE:

La mayor parte de las aplicaciones profesionales se limitan a almacenar, actualizar y presentar con el formato adecuado un conjunto de datos relacionados entre sí. Vienen a ser simplemente una base de datos a la que se ha privado de su flexibilidad.

Si disponemos de una base de datos programable nada nos impide realizar nosotros mismos una aplicación a la medida. Este es el propósito de **Archive:** dotarnos de los medios para construir nuestro programa de gestión.

Las bases de datos, sistemas de tratamiento de información

Todos hemos oído hablar alguna vez de estos programas, pero resulta muy difícil explicar qué se entiende exactamente por este término, ya que el mismo nombre sirve para identificar programas muy diferentes. En esencia, una base de datos es un programa que gestiona la entrada y recuperación de información.

La comparación de estos programas con los archivadores nos ayudará a ver cuál es el avance que supone la informatización de nuestros datos. En un archivo guarda, en cada ficha o carpeta, información que responde a una determinada palabra clave (por ejemplo, en un archivo de facturas, podría ser el nombre de la empresa a la que facturamos). Así, la recuperación de toda nuestra facturación a una determinada empresa equivale a localizar en el archivo ordenado el nombre de la empresa y sacar la carpeta.

El acceso por entradas múltiples a nuestros datos potencia la base de datos respecto al archivador.

Imaginemos, en cambio, que queremos hacer una estadística de las ventas por regiones de un artículo determinado. El problema es mucho más difícil, ya que habrá que repasar empresa por empresa, localizar las facturas que se refieren a ese artículo y clasificarlas por regiones. Un trabajo pesado que nos puede resolver el ordenador, siempre, claro está, que hayamos utilizado una base de datos para almacenar la información.

Más que una base de datos,* Archive *es un verdadero lenguaje de programación, que sirve para resolver cualquier tipo de problema

En esa aplicación, cada factura sería una ficha o registro, donde la información está almacenada en distintos **campos**. Uno podría ser el código de artículo, otro la empresa a la que se factura, etc. En otros archivos auxiliares podemos almacenar las direcciones de las empresas, y en otro una descripción detallada de cada artículo. El programa, para realizar nuestra estadística, selecciona las fichas que se refieren al artículo de interés, y averigua la región en que está la empresa mediante los archivos auxiliares. El trabajo sigue siendo complicado, pero por lo menos parece manejable.

Con este pequeño ejemplo hemos ido subrayando algunas de las características interesantes de una base de datos. Debe permitir trabajar con varios archivos simultáneamente, poder tomar decisiones en función de los valores de los campos, y debe tener funciones de búsqueda y selección que permitan cribar el archivo a nuestro gusto.

Para comenzar, es necesario crear un fichero de datos

Archive es un programa que maneja ficheros de datos. Por tanto, para empezar a trabajar el primer paso es crear un archivo. La instrucción **crear** hace justamente eso. Esta sentencia va seguida del nombre del fichero que almacenará nuestros datos, y en líneas sucesivas van los nombres de los campos que va a tener nuestro archivo, acabados en «$» si son alfanuméricos. Para indicarle al programa que la operación de creación ha acabado basta teclear **fincrear,** aunque si el comando no forma parte de un programa basta teclear una línea vacía.

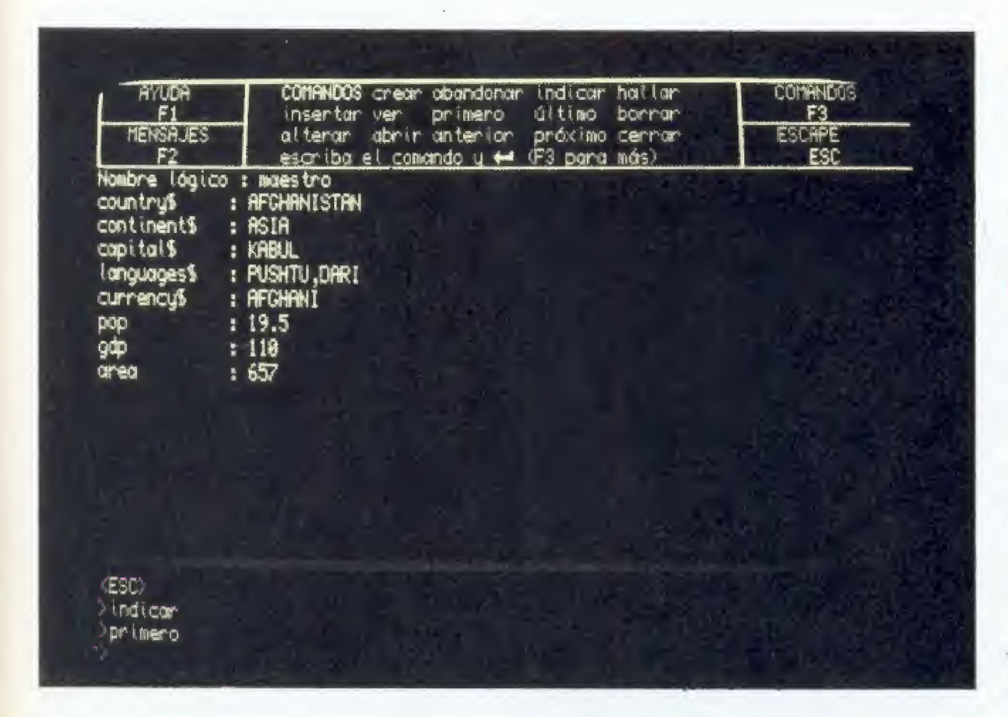
Una vez creado el archivo, la función **cuenta( )** nos devuelve como resultado el número de fichas que contiene nuestro archivo. Si queremos ahora llenar algunas fichas se puede hacer mediante **insertar,** que añade nuevos datos. El programa sitúa el cursor en el primer campo, nos permite introducirlo y editarlo y, al pulsar ENTER o TAB, pasa al campo siguiente. Una pulsación de MAYUSCULAS TAB nos devuelve al campo anterior, que podemos modificar. Mediante F5 se introduce el registro como está en el fichero, y F4 o ESC nos sacan del comando.

Todo el proceso de edición habrá tenido lugar en la pantalla. Como no hemos definido una pantalla de presentación, el programa nos proporciona una bastante ru-

Una Base de Datos a la medida

dimentaria, con los nombres de los campos, uno en cada línea, y espacio para introducir los datos en el resto de la línea. Ese formato se puede modificar fácilmente para presentar la información en pantalla de una manera más cómoda.

Existe un comando muy parecido a éste. Se trata de **Alterar**, que nos enseña el registro sobre el que nos hayamos posicionado y nos deja modificar sus campos. Otra



manera de cambiar registros es utilizando las variables de campo. Estas, que tienen el mismo nombre de la definición del archivo, nos devuelven el campo correspondiente del registro sobre el que estamos posicionados, y se actualizan cada vez que cambiamos de posición. Podemos modificarlas mediante la sentencia **haz** (let), y cuando los valores de los campos sean correctos, bastará introducir **actualiz** para que el programa defina el registro de acuerdo con los valores de las variables de campo. Este procedimiento está pensado para ser utilizado dentro de los programas, y resulta muy cómodo para calcular el valor de un campo en función de otras variables. El comando **añadir** actúa de manera parecida, pero añade un nuevo registro con los valores de las variables de campo. Desde el teclado se puede elegir la ficha de interés recorriéndolo con los comandos **primero, último, próximo** y **anterior,** pero resulta más práctico usar los comandos de búsqueda: **situar, hallar, buscar** y **elegir.** El primero nos sirve para localizar un valor de un campo determinado. Pero para utilizarlo hay que **ordenar** previamente el archivo.

Indexación de archivos. Una herramienta necesaria

La indexación puede incluir hasta cuatro campos, y **situar** puede tener tantos parámetros como campos de ordenación. **Hallar** nos busca una cadena de caracteres en cualquiera de los campos de un archivo. Es el comando a usar cuando no estamos seguros del campo en que aparece la información. Los comandos **buscar** y **elegir** aceptan una expresión lógica. **Buscar** localiza el primer registro que haga la expresión distinta de cero. **Elegir** selecciona los registros que cumplan la condición; a partir de ese momento el fichero se comporta como si no hubiese más registros que los elegidos, y sólo el comando

Una norma a no olvidar: cerrar siempre los archivos

restaurar devuelve el fichero a su condición inicial. Las expresiones se suelen componer de variables de campo.

Por ejemplo, si queremos averiguar cuántos clientes tenemos con saldo negativo, basta teclear **elegir saldo <0.** Después de esta operación, todos los comandos afectarán sólo a los registros con saldo negativo. Si hubiésemos introducido **buscar saldo**<*0,* el programa se detendrá en el primer registro con saldo negativo. Después de tomar medidas contra el moroso, si tecleamos **continuar** el programa irá al siguiente.

Definir pantallas: una buena presentación disminuye errores

Hasta ahora no hemos hablado apenas de los comandos que afectan a la pantalla e impresora. Entre los más importantes están **escribir**, equivalente a PRINT, o **imprimir,** que sustituye a LPRINT. Ambos admiten **en, tab, tinta** o **papel**

ARCHIVE

APLICACION

como parámetros. Otro comando, **vía,** permite que el texto se imprima a **microdrive** o incluso a la pantalla. **Limpiar** borra la pantalla. No olvidemos que, si teníamos definido un formato, será necesario volver a activarlo con **pantalla.**

El editor de pantallas de **Archie** resulta muy útil cuando queremos presentar la salida de una forma agradable a la vista. Una vez introducido el comando **Peditar** se nos permite teclear nuestro «impreso». A continuación, de una manera sencilla, especificamos qué variables se imprimirán y en qué posiciones de la pantalla. El resultado puede ser tan agradable a la vista como queramos y, aunque el editor es poco flexible, es una tarea que se hace sólo una vez.

```
AYUDA     | COMANDOS crear abandonar indicar hallar     | COMANDOS
F1        | insertar ver  primero   último  borrar      | F3
MENSAJES  | alterar  abrir anterior próximo cerrar      | ESCAPE
F2        | escriba el comando y ↵ (F3 para más)        | ESC
Nombre lógico : maestro
country$     : SPAIN
continent$   : EUROPE
capital$     : MADRID
languages$   : SPANISH
currency$    : PESETA
pop          : 36
gdp          : 1966
area         : 505

Error 1:  comando no reconocido
>hallar "españa"
>hallar "spain"
>
```

Los comandos que hemos citado hasta ahora admiten su uso directo desde el teclado o pueden ser incorporados en nuestros programas. Existen instrucciones que sólo tienen sentido dentro de un programa. Una de ellas es **si,** que junto a **sino** y **finsi** nos permiten tomar decisiones. **Mientras** y **finmientras** es una estructura de bucle que itera mientras sea cierta una condición lógica. También resulta muy útil **todos,** que repite las operaciones entre el comando y su correspondiente **finitodos** con todos los registros seleccionados. Si queremos tener varios ficheros abiertos simultáneamente, debemos asignarle un nombre lógico a cada uno.

Los programas de **Archive** se organizan mediante procedimientos, que se pueden llamar como si fueran comandos del sistema. El programa incluye un editor sencillo, pero cómodo de manejar, y una instrucción **traza,** que permite seguir paso a paso la ejecución de nuestros programas.

En resumen, un programa potente, no tan fácil de manejar como sería deseable.

Entre las principales virtudes de **Archive** está su capacidad para ejecutar comandos programados por el usuario, una característica presente sólo en algunos programas para máquinas profesionales, y siempre a un precio poco asequible. Lo peor del programa es el editor de pantalla, algo confuso en su utilización y con el que resulta fácil equivocarse.

Los programas* Archive *se organizan mediante procedimientos

Para hacerle justicia a **Archive** tendríamos que usar **diskettes** y una ampliación de memoria que le dejara «moverse a sus anchas». En general, el programa merece un diez en potencia, un siete en facilidad de uso, un seis en presentación de resultados, y (la versión 2) un ocho en velocidad.

Por último, queremos decir que, dada la extensión de este artículo, es inevitable dejar muchos comandos sin siquiera comentar, y apenas hemos citado un par de funciones. No sería exagerado decir que **Archive** es un lenguaje de programación, y que su potencia permite escribir programas para cualquier tipo de aplicación. El programa no resulta difícil de manejar, pero si queremos «exprimir» las posibilidades de la máquina, habrá que practicar bastante. ¡Por cierto! Si algún lector ha escrito alguna aplicación interesante, estaremos encantados de publicarla en nuestras páginas.



Cartridge Doctor, para no perder información

Los programas para recuperar información de discos parcialmente estropeados son de uso cotidiano en máquinas como el **Apple II** o el **IBM PC.** Teniendo en cuenta la fragilidad del *microdrive* era raro que ninguna compañía inglesa se hubiera lanzado hasta ahora a escribir un buen programa para cuidar nuestros cartuchos. **Serma** se ha traído de Gran Bretaña este doctor, escrito por los chicos de **Talent.**

Según el manual de instrucciones, que resultará claro para quienes dominen el inglés, el programa utiliza avanzados conceptos de inteligencia artificial (¿?). El programa parece «listillo», pero no tan inteligente como nos quieren hacer creer sus autores.

Nada más arrancar, disponemos de cinco opciones: AUTO-CLONE, que duplica un cartucho en otro, incluyendo los ficheros que habían sido borrados y aquellos que dan problemas de carga. FILE PATCH nos permite editar archivos al nivel más bajo, parchear aquellos ficheros que nos den problemas de lectura, o simplemente curiosear su contenido.

SALVAGE SECTIONS sirve para copiar parte de un archivo a otro, con la posibilidad de saltar una parte defectuosa. La cuarta opción nos ofrece un catálogo ampliado, con el número de bloques que ocupa el archivo, y qué entrada ocupa en el directorio. Por último, TRANSLITERATE copia un archivo en otro cambiando todas las ocurrencias de un caracter dado por otro.

Salvar cartuchos dañados, una tarea fácil

La opción de duplicación es la primera que debe usarse cuando tenemos problemas de carga; así evitamos dañar aún más el cartucho en nuestros torpes intentos de salvación. Para ello es necesario disponer de un cartucho vacío, en el que se realizará la copia. El programa empieza mirando si el directorio y el mapa de sectores están en buen estado. A continuacion nos presenta el nombre de cada fichero, con el mensaje OK si lo ha leído bien, y el número de bloques incorrectos en caso contrario. Se nos da la opción de pasar cada fichero al nuevo cartucho.

Una vez acabada la copia podemos optar por intentar parchear los archivos que no hayan podido ser recuperados correctamente. Si éstos están en código máquina, los resultados serán negativos la mayor parte de las veces. Si son BASIC o de texto, aún podemos conservar esperanzas.

No os sorprendais si el programa presenta en pantalla nombres de archivos que borrásteis hace meses: una de sus características es que recupera ficheros borrados, siempre que el espacio que ocupaban no haya sido «machacado» por otro archivo. Así, si accidentalmente perdemos algún dato importante, no debemos volver a utilizar el cartucho hasta haberlo recuperado mediante **Cartridge Doctor**.

La opción de parcheado es un editor muy simple, que nos presenta caracter a caracter cada bloque, pudiendo escribir en él, pasar al bloque siguiente, etc. Si, por ejemplo, un bloque es leído con dificultad, basta situarse en él (tras una buena lectura) y decirle al programa que lo vuelva a escribir. Generalmente desaparecerán los problemas, y la carga será más rápida.

Sólo para copias de seguridad

Más de un pirata se frotará las manos al leer este artículo, pero mucho nos tememos que sus ilusiones ser verán defraudadas: **Cartridge Doctor** sólo copia, no desprotege. Cualquier programa protegido mantendrá su protección, y los programas que no lo están se pueden copiar desde el BASIC, así que no se trata de «abrir». Este programa es mucho más útil: recupera información que se quedó enredada en los bucles del *microdrive*.



CUB Y QL LA PAREJA PERFECTA

Este monitor en color CUB es TOTALMENTE compatible con el Sinclair QL.
Combina las capacidades de gráficos destacados con la facilidad de visualizar textos en 85 columnas mientras hace justicia al potencial de color del QL. Su nuevo diseño dotado de una base basculante y giratoria armoniza perfectamente con la simple apariencia funcional del QL.
Lo mejor de todo, es su precio, solo de

91.600

P.V.P., incluyendo su nueva base.

ESPECIFICACIONES

MODELO-CUB
1451/DQT3
MONITOR DE 14"
ENTRADA - RGB/TTL
RESOLUCION - 653 (H) × 585 (V)
DOT PICH - 0,43 mm.
ANCHO DE BANDA - 18 MHz
ANTIREFLEJO CRT





Pº de la Habana, 145
Telf. 458 74 75 - 28036 Madrid

Aprendiendo Lenguaje Máquina

CAPITULO: 2

La memoria

Como su propio nombre indica, esta vale para almacenar los datos y programas que se ejecutan, sin ella el ordenador sería un amnésico perdido y no serviría para nada al no poder recordar nada. Es más, debido a que todos los programas que ejecuta se almacenan en memoria, se podría decir que un ordenador sin memoria está en coma profundo ya que no sería capaz de hacer nada (ni siquiera nos podría poner en pantalla el mensaje inicial de Sinclair Research ya que esto también lo hace un programa especial denominado «sistema operativo»).

Esta memoria se divide en dos grupos distintos. El primero es el compuesto por la «memoria ROM» (abreviatura de «Read Only Memory»), ésta es de un tipo que sólo se puede leer por microprocesador y no escribir. Esto puede crear la idea de que entonces no tiene ninguna utilidad, ya que no se pueden introducir datos. Pero la realidad nos demuestra que hay una serie de programas que nunca se modifican y que vienen con la máquina, por lo que es más conveniente este sistema que no leerlos de cinta cada vez que los necesitemos. Entre otros programas hay que destacar al sistema operativo y al intérprete BASIC. Estos se graban en la ROM directamente en la fábrica por procesos especiales.

La RAM (nombre que proviene del término «Random Access Memory») consiste en la memoria en la que sí se puede escribir además de leer. En ella se almacenan los datos, los programas que tecleamos o cargamos de cinta, etc. Este tipo de memoria es, por tanto, modificable y todos los datos guardados en ella se borran al apagar la máquina, perdiéndose irremediablemente si no los hemos grabado en cinta.



El funcionamiento de la memoria dentro del ordenador es un tanto especial, por lo que lo examinaremos más atentamente. Como comentábamos antes, cada memoria individual consiste en una casilla en la que se pueden almacenar datos, cada una de estas casillas está numerada y no hay dos que posean el mismo número, ya que entonces nuestro «cartero», el Z-80, no sabría a cuál de las dos dirigirse. La cantidad de casillas disponibles está limitada, por tanto, por los números que se puedan usar y esto lo determina el microprocesador. En el Z-80, como veremos posteriormente, el número inferior que



se puede utilizar es el 0, y el mayor el 65535(FFFFh), por lo que existen 65536 casillas distintas.

Otro detalle característico de las memorias en su almacenamiento, cada vez que introducimos un dato en una de ellas, el que existiese en esa misma previamente se borra y es sustituido por el nuevo. Por tanto, cada una de ellas sólo puede contener un número que debe estar comprendido entre 0 y 255 (un *byte* que es lo que cabe en cada una). También peculiar es el caso de sacar un dato de una casilla para leerlo, ya que, al contrario que las cartas del buzón, cuando sacamos un dato de una memoria, sigue existiendo una copia de él en la memoria. En realidad el único método existente para borrar un dato de una memoria es escribir otro encima. (Por supuesto este método de escritura sólo se puede usar con la memoria RAM y no con la ROM).

Los dispositivos de entrada y salida

Un proceso vital que debe realizar todo ordenador para ser útil es el de comunicarse con el mundo que le rodea. Esta frase tan grandilocuente se transforma en el Spectrum en el proceso de leer datos del teclado, mostrarlos por pantalla, grabar y leer en cinta, etc. Resulta evidente que si no se pudiesen realizar estos procesos el ordenador sería un trasto totalmente inútil ya que aunque fuese muy potente, nosotros no podríamos utilizarlo para nada.

Estas funciones las realiza en el Spectrum el microprocesador ayudado por un *chip* llamado «ULA» (Uncomited Logic Array) que es lo que podríamos denominar un «procesador de entrada y salida»; es decir, una especie de microprocesador diseñado específicamente para estas funciones.

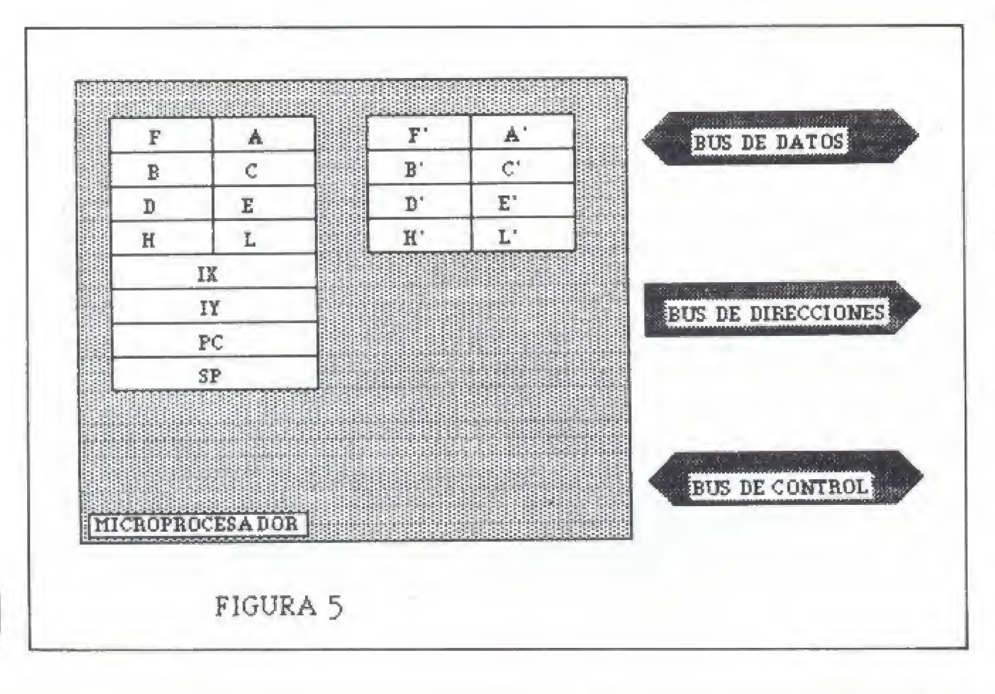


FIGURA 5

El microprocesador

Este es el famoso Z-80 y fue diseñado por la compañía norteamericana Zilog (la mayoría de los componentes de un ordenador no son construidos por el fabricante de éste, sino que los compra a otras empresas especializadas en estos trabajos). Este *chip* (o cucaracha, como se le denomina vulgarmente en España) se encarga de controlar todos los procesos y de realizar todos los cálculos. Una representación por bloques de su estructura se puede ver en la figura 5. En ella vemos que dentro de él existen una serie de «cajas» denominadas registros, en los cuales se almacenan diversos datos necesarios para el funcionamiento y los cálculos que le ordenemos. También existen tres *buses* que salen de él hacia el exterior. Estos son una serie de líneas por las que se controlan las memorias y el procesador de entrada y salida.

En el *bus* de direcciones se envían los números empleados para seleccionar una memoria u otra, de modo que si aquí hay un cero, se manejará la memoria cero y se hay un mil, se manejará la memoria mil. El *bus* de datos es por donde se envían y reciben los datos a las memorias y al procesador de entrada y salida.

Por último existe un *bus* de control que se encarga de llevar todas las demás señales necesarias para el correcto funcionamiento de la máquina y que iremos viendo posteriormente, interrupciones, control de la ULA, refresco de la memoria, etcétera.

Los registros internos que comentamos antes tienen cada uno encomendadas misiones específicas como vamos a ver.

El primero es el acumulador, representado por la letra «A». Aquí se realizan todas las operaciones y se guardan los resultados. Su equivalente más directo es la pantalla de una calculadora, en la que se





meten todos los datos y donde se muestran los resultados. Su capacidad es la misma que la de las memorias, un *byte*. En realidad se debe a la capacidad de este registro el que las memorias no puedan contener más datos cada una, ya que aunque las hiciésemos más grandes, no cabrían aquí y los dígitos sobrantes se desperdiciarían.

El registro PC es, junto con el acumulador, el más importante del microprocesador ya que es el contador del programa, es decir, aquí se almacena la dirección de la memoria donde está almacenada la siguiente instrucción del programa que se está ejecutando. Es un registro de 16 *bits*, por lo que ésta es la cantidad máxima de memoria que puede direccionar el microprocesador, cualquier memoria situada fuera de este rango no podría ser manejada y sería inútil. El manejo de este registro lo realiza, casi siempre, el microprocesador por su cuenta y lo incrementa después de cargar una instrucción para que apunte a la siguiente de modo que cuando la tenga que ejecutar ya sepa cuál es su dirección, que se saca por el *bus* de direcciones.

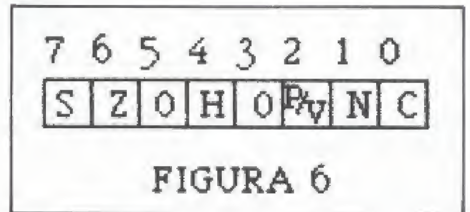


FIGURA 6

El segundo registro de 16 *bits* existente en el Z-80 es el «SP», denominado *stack pointer*, que traducido al español significa «puntero de la pila». Esta es una zona de la memoria general que se reserva para ser usada como una memoria «LIFO», que resulta muy útil para almacenar y sacar datos a gran velocidad, aunque debido a las restricciones de este manejo (sólo se

01 001 010



puede manejar la última que se ha metido) no se puede usar siempre.

Otros registros son los formados por las parejas B-C, D-E y H-L. Cada una de estas parejas se puede usar como un registro de 16 *bits* para apuntar a direcciones de memoria, como dos de 8 *bits* en los que se guardan datos de uso general. Estos registros son muy versátiles y aunque no permiten realizar operaciones aritméticas en ellos como se hace con el acumulador, son muy usados.

Los registros IX e IY son registros de indexación. Este es un sistema para usar la memoria que es muy útil, sobre todo, para hacer contadores o manejo de matrices desde lenguaje máquina. Estos dos registros se usan siempre como de 16 *bits* ya que están destinados a direccionar la memoria.



Las banderas internas

A continuación tenemos un registro F. Este contiene un *byte*, pero aquí no expresa un número, sino que cada uno de sus *bits* tiene una función independiente de los demás; en este caso a cada *bit*, debido a que tiene una función distinta, se le denomina bandera. Cada una de ellas es similar a un interruptor de la luz que cuando está «encendido» indica una cosa y cuando está «apagado» la contraria. La disposición de estas dentro del *byte* es la que se indica en la figura 6. A continuación vamos a ver las funciones básicas de cada una de ellas, aunque posteriormente se explicarán algunos usos no tan específicos como estos.

La primera que vamos a tratar es la bandera de acarreo («C»). Esta es modificada por un gran número de operaciones matemáticas y lógicas. Quizá la principal de ellas, por lo menos por la que recibió el nombre, sea la suma. Recordarán que al hablar de la suma de números binarios, vimos cómo se hacía y que en muchos casos, como en cualquier otro sistema de numeración, resultaba que «nos llevábamos uno» al sumar. Pues bien, si trabajamos con *bytes* pueden darse casos en los que obtengamos un resultado con 9 *bits*, como por ejemplo «1001000b» más «0111010b», que da «100000110b». Siendo el *bit* más alto (el octavo ya que empezamos a contar desde el cero) un uno. Este no nos cabe en el acumulador porque sólo tiene capacidad para un *byte* (8 *bits*), lo que hace el microprocesador es meter los 8 más bajos en éste y el noveno lo mete en la bandera de acarreo. De





modo que si la suma da más de 8 *bits* se mete un uno, y si no un cero. En cierto modo se puede considerar una extensión del acumulador para ciertas operaciones pero se tiene que recordar que si leemos o grabamos este en alguna memoria, *no* se grabará el acarreo.

La bandera N es usada internamente por el microprocesador para indicar si se ha realizado una suma o una resta y no suele tener utilidad para el programador.

La bandera «P/V» tiene dos funciones distintas. Cuando se realiza una operación aritmética en el acumulador con signo (es decir, se utilizan sólo siete *bits* para expresar el número mientras que el último indica el signo, un uno indica negativo y un cero positivo) éste indica si se ha producido un error de exceso en el cálculo. Es, por tanto, similar a la de acarreo pero en ésta se consideraba el número sin signo. El modo de realizar operaciones con signo lo veremos más adelante, ahora nos basta con saber que existe.

Otro tipo de acarreo que toma en consideración el microprocesador es el del *nibble*, para lo cual se usa la bandera «H» como la «C» pero en lugar de indicar un exceso en la suma de todo el *byte*, todo el *nibble* inferior y se pone a uno si al sumar da un 1 en el *byte* siguiente al del *nibble*. Es decir, al realizar sumas aquí se almacena el *byte* 5 del acumulador, pero con las restas puede variar. En general se usa en las sumas de números pequeños o en ocasiones especiales.

La bandera «Z» indica cuándo en el acumulador hay un cero. Esto es muy útil cuando se realizan bucles que se terminan al llegar determinado valor a cero, ya que basta comprobar esta bandera y proceder en consecuencia.





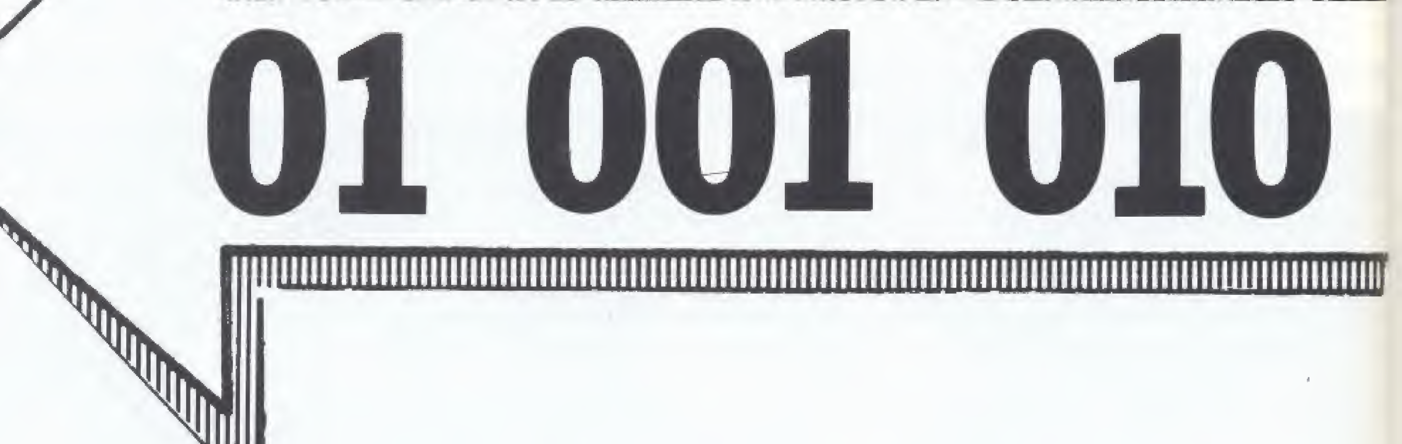
Al igual que la anterior indicaba cuándo había un cero en el acumulador, la «S» indica con un uno que hay un número negativo y con un cero que hay un positivo. Esto, evidentemente, sólo tiene sentido cuando se opera con números con signo. Esta bandera en realidad adopta el mismo valor que el *bit* más alto del acumulador (el 7).

Las dos banderas que en la figura están a cero no se usan y su valor resulta indiferente para nosotros.

Los registros paralelos

Al ver cómo eran los registros internos del acumulador, vimos que existía un grupo compuesto por A', F', B', C', D', E', H' y L' que eran paralelos a los que hemos visto. Estos registros primos forman un grupo especial que no se puede utilizar normalmente y que se cambian con los principales mediante una instrucción especial en cuyo caso sólo podemos usar los nuevos. Su uso se suele reservar para ejecutar rutinas a gran velocidad para evitar perder tiempo guardando los valores actuales y recuperando los nuevos, ya que con una sola instrucción se cambian entre sí.

Fernando García





Guía del comprador de Todospectrum

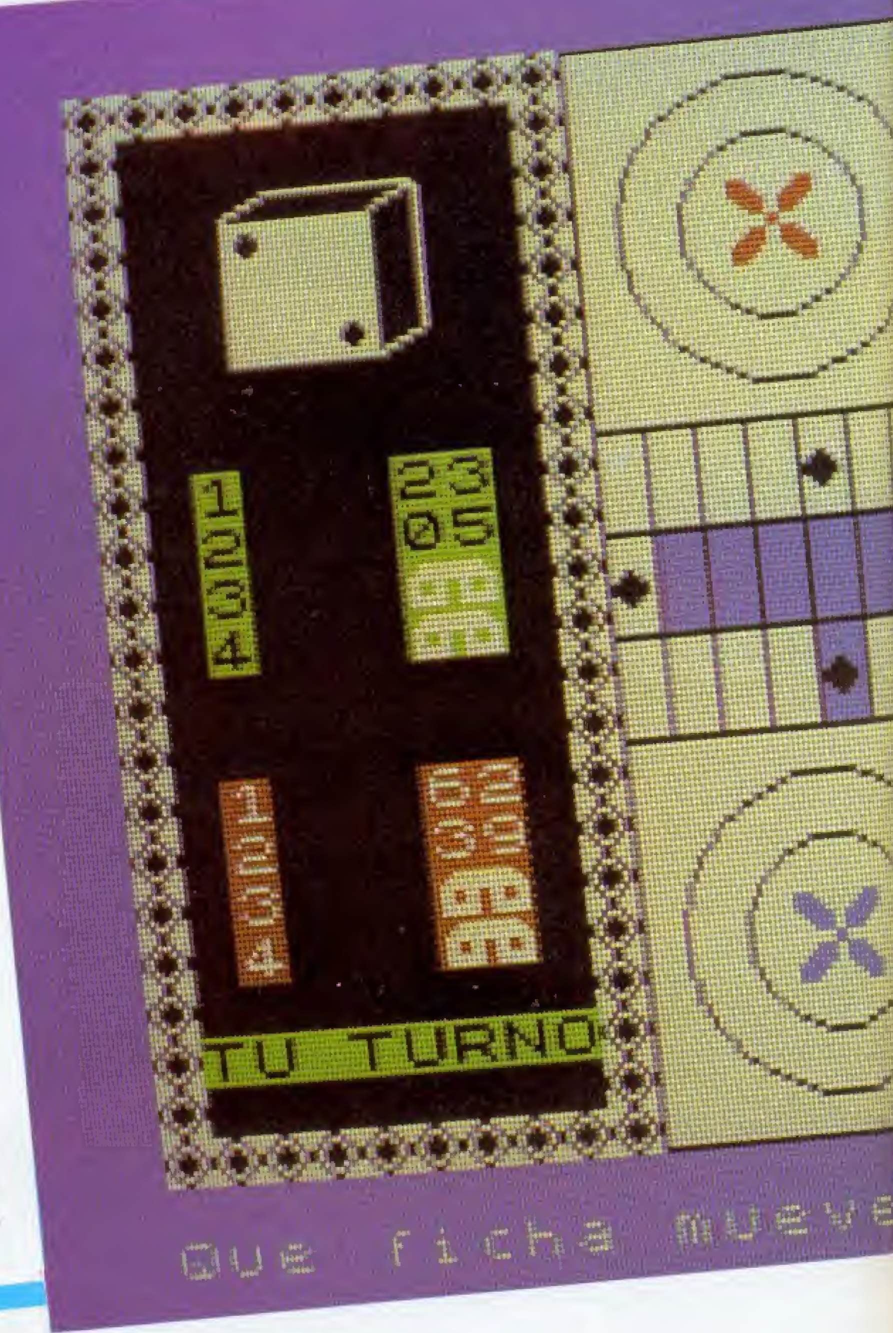
Parchis

Aunque todavía disfrutamos de buena temperatura, y el otoño sólo ha amenazado, llega el momento de ser previsor, y prepararse para las largas y aburridas noches de invierno. Para ellas Raúl González nos envía un programa: ¿Qué mejor que un juego clásico como el parchís? Y, si no tienes compañeros para jugar, tranquilo: tu Spectrum está siempre dispuesto.

El listado que os presentamos permite jugar una partida de parchís contra el ordenador. Este tira los dados y mueve sus fichas; después del movimiento, tira los dados y pregunta qué ficha quieres mover. No trates de engañarle, ya que se sabe muy bien las reglas.

En la primera jugada el Spectrum tira los dados para ti; si sacas un cinco te coloca dos fichas en la casilla de salida. Si es cualquier otra puntuación sacará una ficha y moverá tantos puntos como hayamos sacado. En esto se diferencia un tanto de las reglas a las que estamos acostumbrados. Por lo demás, las reglas son bastante usuales.

Para sacar una pieza nos hace falta un cinco. Las barreras no se pueden pasar, ni se puede salir del juego si no es con la puntuación justa. Cuando comemos podemos elegir con qué pieza recorremos la recompensa, etc.




```
   1 BORDER 1: CLS
   2 PRINT #0;"PARCHIS PARCHIS
PARCHIS PARCHIS"
   3 RANDOMIZE
   6
   7
   8
  15 REM ____________________
  16
  17
  18 REM   RAUL GONZALEZ BRAVO
  19
  20
  21 REM ____________________
  30
```



Si tenemos todas las fichas y sacamos un seis, movemos siete casillas, además de volver a tirar. Si repetimos tres veces el seis, nos vamos directamente a casa.

Como en otras ocasiones, las letras que aparecen subrayadas en el listado corresponden a los caracteres gráficos correspondientes del Spectrum. Los triángulos que aparecen en algunas ocasiones detrás de PRINT corresponden al sostenido (también llamado «almohadilla»). Esperamos perdonéis estas excentricidades de nuestra impresora, que se quiere hacer notar.

El único inconveniente que puede presentar el programa es que se trata de un juego solitario: tu contra el ordenador. Si alguien se atreve a adaptar el juego para que admita varios «humanos», o el Spectrum juegue varios turnos, ya sabe: tiene un hueco en nuestra sección de programas. Por lo demás, los gráficos son excelentes, y los tiempos de respuesta son buenos. Para jugar basta tener la paciencia de introducirlo y un Spectrum de 48 K.

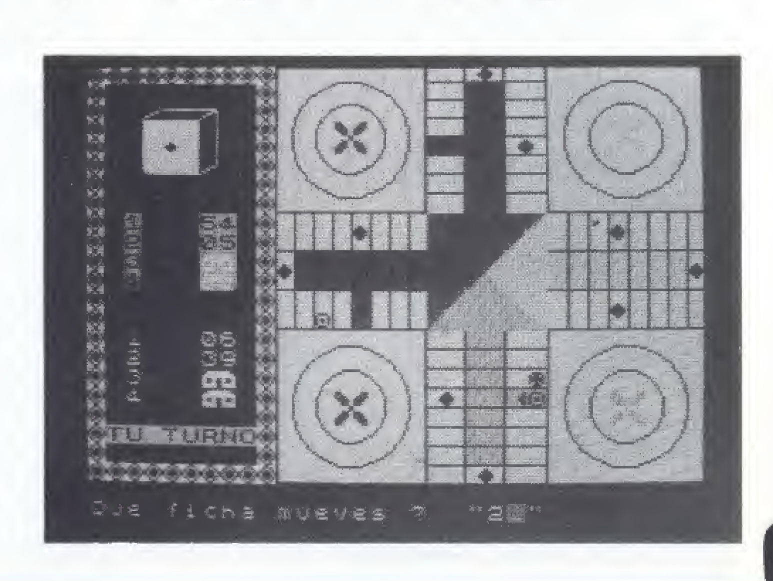
Raúl González Bravo **48 K**

```
   31
   32
   35 GO SUB 230
   40 LET z$=" ": LET s$="          B
ienvenido  al juego del parchis,
 vamos a enfrentarnos tu y yo  e
n  este apasionante juego.
                       Estas son tu
s fichas:   G      Estas son las
mias   :   B"
   50 LET s$=s$+"           Tus  fic
has son de color verde estando s
ituada tu casa abajo a la derech
a y las mias son rojas  y  estan
 arriba  a la izquierda.         "
   60 LET s$=s$+"                   E
n la primera tirada  yo te muevo
 las fichas; si sacas un cinco t
e saldran  dos fi- chas, si es
otro numero colo- care  una fich
a en la casilla a partir de la s
alida (num.5)"
```

Programas

```
 139 LET d=1: LET f=2
 141 FOR n=1 TO LEN s$
 142 IF s$(n)<>z$ THEN  BEEP .05
,INT (RND*7)
 143 PRINT AT f,d;s$(n)
 145 LET d=d+1
 146 IF d=31 THEN  LET d=1: LET
f=f+1
 147 NEXT n
 148 PRINT AT 21,1;"PULSA UNA TE
CLA PARA CONTINUAR": PAUSE 0
 149 FOR n=1 TO 22: RANDOMIZE US
R 3190: NEXT n
 150 PRINT  FLASH 1;AT 10,8;"ESP
ERA UN MOMENTO"
 151 LET cosiet=0: LET cosies=0:
 LET comiet=0: LET comies=0: LET
 casass=0: LET casat=0: LET seis
t=0: LET seiss=0
 152 LET gans=0: LET gant=0: LET
 perss=0: LET perst=0
 153 DIM k(50): DIM r(4): DIM z(
12): DIM c(100): DIM x(86,2): DI
M y(86,2)
 154 RESTORE 159
 155 FOR n=1 TO 48
 156 READ a
 157 LET k(n)=a
 158 NEXT n
 159 DATA 12,17,51,56,22,46,12,1
7,51,56,22,46,5,29,34,39,63,68,5
,29,34,29,63,68
 160 DATA 1,2,1,1,2,2,2,1,2,2,1,
1,2,2,1,1,1,2,1,1,2,2,2,1
 161 RESTORE 166
 162 FOR n=1 TO 86
 163 READ a,b
 164 LET x(n,1)=a: LET y(n,1)=b
 165 NEXT n
 166 DATA 26,16,28,16,26,18,28,1
8,23,21,23,20,23,19,23,18,23,17,
23,16,23,15,23,14,24,13,25,13
 167 DATA 26,13,27,13,28,13,29,1
3,30,13,31,13,31,10,31,8,30,8,29
,8
 168 DATA 28,8,27,8,26,8,25,8,24
,8,23,7,23,6,23,5,23,4,23,3,23,2
,23,1,23,0
 169 DATA 20,0,18,0,18,1,18,2,18
,3,18,4,18,5,18,6,18,7,17,8,16,8
,15,8,14,8,13,8
 170 DATA 12,8,11,8,10,8,10,11,1
0,13,11,13,12,13,13,13,14,13,15,
13,16,13,17,13,18,14
 171 DATA 18,15,18,16,18,17,18,1
8,18,19,18,20,18,21,21,21,21,20,
21,19,21,18,21,17,21,16,21,15,21
,14
 172 DATA 21,1,21,2,21,3,21,4,21
,5,21,6,21,7
 173 FOR n=1 TO 86
 174 READ a,b
 175 LET x(n,2)=a: LET y(n,2)=b
 176 NEXT n
 177 DATA 12,2,15,2,12,5,15,5
 178 DATA 22,21,22,20,22,19,22,1
8,22,17,22,16,22,15,22,14,24,12,
25,12,26,12,27,12,28,12,29,12
 179 DATA 30,12,31,12,31,11,31,9
,30,9,29,9,28,9,27,9,26,9,25,9,2
4,9,22,7,22,6,22,5,22,4,22,3,22,
2
 180 DATA 22,1,22,0,21,0,19,0,19
,1,19,2,19,3,19,4,19,5,19,6,19,7
,17,9,16,9,15,9,14,9,13,9,12,9
 181 DATA 11,9,10,9,10,10,10,12,
11,12,12,12,13,12,14,12,15,12,16
,12,17,12,19,14,19,15,19,16,19,1
7,19,18,19,19,19,20,19,21,20,21
 182 DATA 20,20,20,19,20,18,20,1
7,20,16,20,15,20,14,20,1,20,2,20
,3,20,4,20,5,20,6,20,7
 183 RESTORE 188
 184 FOR n=1 TO 12
 185 READ a
 186 LET z(n)=a
 187 NEXT n
 188 DATA 5,12,17,22,29,34,39,46
,51,56,63,68
 200 GO TO 300
 230 RESTORE 250
 240 FOR n=1 TO 18: READ a$: FOR
 s=0 TO 7: READ a: POKE USR a$+s
,a: NEXT s: NEXT n
 250 DATA "a",1,3,7,15,31,63,127
,255
 252 DATA "b",255,127,63,31,15,7
,3,1
```

```
 254 DATA "c",0,0,24,60,60,24,0,
0
 256 DATA "d",24,36,90,189,189,9
0,36,24
 258 DATA "e",0,1,3,7,7,3,1,0
 260 DATA "f",0,128,192,224,224,
192,128,0
 262 DATA "g",0,0,0,0,0,24,60,12
6
 264 DATA "h",126,60,24,0,0,0,0,
0
 266 DATA "i",0,112,120,124,62,3
0,14,1
 268 DATA "j",0,14,30,62,124,120
,112,128
 270 DATA "k",1,14,30,62,124,120
,112,0
 272 DATA "l",128,112,120,124,62
,30,14,0
 274 DATA "m",0,31,63,127,102,10
2,126,126
 276 DATA "n",0,248,252,254,102,
102,126,126
 278 DATA "o",0,123,65,65,113,65
,67,0
 280 DATA "p",0,162,50,42,38,34,
162,0
 282 DATA "q",0,60,66,90,90,66,6
0,0
 284 DATA "r",0,60,90,126,126,90
,60,0
 290 RETURN
 300 CLS
 380 FOR s=0 TO 9 STEP 9: FOR n=
0 TO 21: PRINT AT n,s;"D": NEXT
n: NEXT s
 385 FOR n=0 TO 21 STEP 21: PRIN
T AT n,1;"DDDDDDDD": NEXT n
 390 FOR n=1 TO 20: PRINT AT n,1
;"[redacted]": NEXT n
 400 FOR n=1 TO 7: PRINT  PAPER
2;AT n,20;"  ": NEXT n
 410 FOR n=14 TO 20: PRINT  PAPE
R 4;AT n,20;"  ": NEXT n
 420 FOR n=10 TO 11: PRINT  PAPE
R 1;AT n,11;"       ": NEXT n
 430 FOR n=10 TO 11: PRINT  PAPE
R 6;AT n,24;"       ": NEXT n
 440 INK 1: LET eme=0: FOR n=144
 TO 166: PLOT n,66+eme: DRAW 0,4
4-(2*eme): LET eme=eme+1: NEXT n
 450 INK 6: LET eme=0: FOR n=190
 TO 168 STEP -1: PLOT n,66+eme:
DRAW 0,44-(2*eme): LET eme=eme+1
: NEXT n
 460 INK 4: LET eme=0: FOR n=64
TO 80: PLOT 146+eme,n: DRAW 44-(
2*eme),0: LET eme=eme+1: NEXT n
 470 INK 2: LET eme=0: FOR n=111
 TO 90 STEP -1: PLOT 145+eme,n:
DRAW 44-(2*eme),0: LET eme=eme+1
: NEXT n
 500 LET h=18: FOR n=13 TO 11 ST
EP -1: PRINT  PAPER 1; INK 4;AT
n,h;"A": LET h=h+1: NEXT n
 502 LET h=23: FOR n=13 TO 11 ST
EP -1: PRINT  PAPER 4; INK 6;AT
n,h;"B": LET h=h-1: NEXT n
 504 LET h=18: FOR n=8 TO 10: PR
INT  PAPER 1; INK 2;AT n,h;"B":
LET h=h+1: NEXT n
 506 LET h=23: FOR n=8 TO 10: PR
INT  PAPER 2; INK 6;AT n,h;"A":
LET h=h-1: NEXT n
 508 INK 0:
 510 RESTORE 530
 512 FOR n=1 TO 6
 514 READ a,b: PRINT  OVER 1;AT
a,b;"H"
 516 PRINT  OVER 1;AT a-1,b;"G"
 520 IF z(n)=22 THEN  PRINT  PAP
ER 6;AT 9,27;"H";AT 8,27;"G"
 524 IF n=6 THEN  PRINT  PAPER 1
;AT a,b;"H";AT a-1,b;"G"
```



Programas

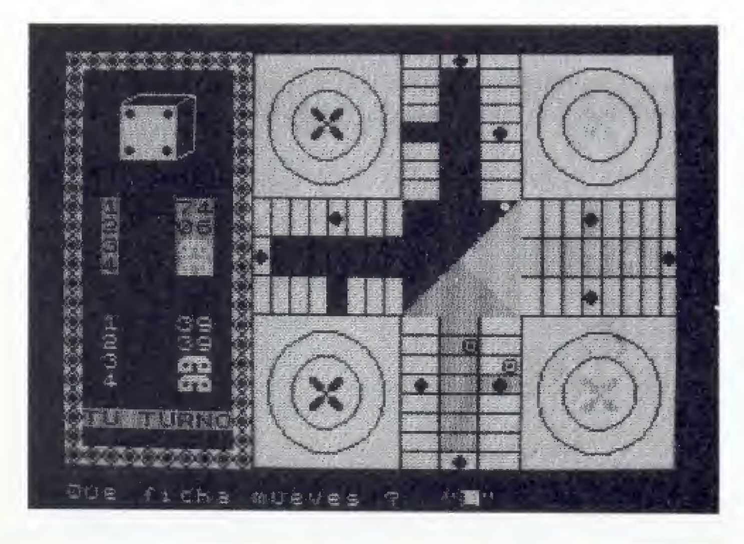
```
 526 NEXT n
 530 DATA 13,27,11,31,9,27
 532 DATA 9,14,11,10,13,14
 536 RESTORE 550
 540 FOR n=1 TO 6
 542 READ a,b: PRINT  OVER 1;AT
a,b;"EF"
 544 IF n=6 THEN  PRINT  PAPER 4
;AT a,b;"EF"
 546 IF n=3 THEN  PRINT  PAPER 2
;AT a,b;"EF"
 548 NEXT n
 550 DATA 4,22,0,20,4,18,17,18,2
1,20,17,22
 560 GO TO 800
 600 FOR n=0 TO 56 STEP 8: PLOT
144,n: DRAW 48,0: NEXT n
 610 FOR n=120 TO 175 STEP 8: PL
OT 144,n: DRAW 48,0: NEXT n
 620 FOR n=80 TO 136 STEP 8: PLO
T n,64: DRAW 0,48: NEXT n
 630 FOR n=200 TO 255 STEP 8: PL
OT n,64: DRAW 0,48: NEXT n
 640 PLOT 143,0: DRAW 0,63: PLOT
 192,0: DRAW 0,63
 644 NEXT n
 645 PLOT 143,175: DRAW 0,-63: P
LOT 192,175: DRAW 0,-63
 650 PLOT 80,64: DRAW 63,0: PLOT
 80,112: DRAW 63,0
 655 PLOT 192,64: DRAW 63,0: PLO
T 192,112: DRAW 63,0
 660 PLOT 80,0: DRAW 0,175: DRAW
 175,0: DRAW 0,-175: DRAW -175,0
 670 PLOT 255,80: DRAW -63,0: PL
OT 255,96: DRAW -63,0
 680 PLOT 80,80: DRAW 63,0: PLOT
 80,96: DRAW 63,0
 690 PLOT 159,0: DRAW 0,63: PLOT
 175,0: DRAW 0,63
 700 PLOT 159,175: DRAW 0,-63: P
LOT 175,175: DRAW 0,-63
 710 RETURN
 800 FOR n=15 TO 25 STEP 10: CIR
CLE 111,32,n: NEXT n
 805 FOR n=15 TO 25 STEP 10: CIR
CLE 223,32,n: NEXT n
 810 FOR n=15 TO 25 STEP 10: CIR
CLE 223,143,n: NEXT n
 815 FOR n=15 TO 25 STEP 10: CIR
CLE 111,143,n: NEXT n
 820 PRINT  INK 1;AT 17,13;"IJ";
AT 18,13;"KL"
 825 PRINT  INK 4;AT 17,27;"IJ";
AT 18,27;"KL"
 830 PRINT  INK 6;AT 3,27;"IJ";A
T 4,27;"KL"
 835 PRINT  INK 2;AT 3,13;"IJ";A
T 4,13;"KL"
 900 FOR n=1 TO 4: PRINT  PAPER
4; INK 0;AT 7+n,2;n: PRINT  PAPE
R 4; INK 7;AT 7+n,6;"MN": NEXT n
 950 FOR n=1 TO 4: PRINT  PAPER
2; INK 7;AT 13+n,2;n: PRINT  PAP
ER 2; INK 7;AT 13+n,6;"MN": NEXT
 n
 960 GO SUB 600
1000 INK 0
1010 DIM s(4): DIM t(4)
1020 FOR n=1 TO 4: LET t(n)=n: L
ET s(n)=n: NEXT n
2000 REM TIRADA
2001 OVER 1: PLOT 24,152: DRAW 7
,4: DRAW 24,0: DRAW 0,-24: DRAW
-7,-4: PLOT 48,152: DRAW 7,4: OV
ER 0
2004 LET tur=0
2008 IF tur=0 THEN  PRINT  PAPER
 4;AT 19,1;"TU TURNO"
2009 IF tur=1 THEN  PRINT  PAPER
 2;AT 19,1;"MI TURNO"
2015 LET cant=INT (RND*4)+7
2020 FOR n=1 TO cant
2021 BEEP 0.03,10
2030 LET tir=INT (RND*6)+1
2040 GO SUB 9000+(10*tir)
2050 PAUSE 20
2060 NEXT n
2061 BEEP 0.5,-20
2065 IF cosiet=1 THEN  GO TO 300
0
2067 IF cosies=1 THEN  GO TO 700
0
2070 IF comiet=0 THEN  GO TO 250
0
2080 IF comies=0 THEN  GO TO 270
0
2100 IF tur=0 THEN  GO TO 3000
2110 IF tur=1 THEN  GO TO 7000
```

```
2500 REM PRIMERA TIRADA JUGADOR
2510 LET comiet=1
2520 IF tir<>5 THEN  LET t(1)=9+
tir: LET tur=1: GO TO 2600
2530 IF tir=5 THEN  LET t(1)=9:
LET t(2)=9: LET tur=1: GO TO 265
0
2600 PRINT AT y(t(1),1),x(t(1),1
);"@"
2605 GO SUB 600
2610 LET c(t(1))=11
2620 IF (t(1)-4)<10 THEN  PRINT
 PAPER 4; INK 0;AT 8,6;"0";(t(1)
-4)
2630 IF (t(1)-4)>=10 THEN  PRINT
  PAPER 4; INK 0;AT 8,6;(t(1)-4)
2635 IF tir=6 THEN  LET seist=1:
 LET cosiet=1: GO TO 2004
2636 LET casat=1
2640 GO TO 2005
2650 PRINT  PAPER 4;AT y(t(1),1)
,x(t(1),1);"@";AT y(t(2),2),x(t(
2),2);"@"
2655 GO SUB 600
2660 LET c(t(1))=13
2670 PRINT  PAPER 4; INK 0;AT 8,
6;"0";(t(1)-4)
2675 PRINT  PAPER 4; INK 0;AT 9,
6;"0";(t(2)-4)
2680 LET casat=2: GO TO 2005
2700 REM PRIMERA TIRADA SPECTRUM
2710 LET comies=1
2720 IF tir<>5 THEN  LET s(1)=43
+tir: GO TO 2800
2730 IF tir=5 THEN  LET s(1)=43:
 LET s(2)=43: GO TO 2900
2800 PRINT AT y(s(1),1),x(s(1),1
);"R"
2805 GO SUB 600
2810 LET c(s(1))=1
2815 LET r(1)=1
2820 PRINT  PAPER 2; INK 7;AT 14
,2;"1";AT 14,6;(s(1)-4)
2825 INK 0
2830 IF tir=6 THEN  LET seiss=1:
 LET cosies=1: LET tur=1: GO TO
2005
2850 LET casass=1: GO TO 2004
2900 PRINT  PAPER 2; INK 0;AT y(
s(1),1),x(s(1),1);"R";AT y(s(2),
2),x(s(2),2);"R"
2905 GO SUB 600
2910 LET c(s(1))=3
2916 LET r(1)=1: LET r(2)=1
2918 PRINT  PAPER 2; INK 7;AT 14
,6;(s(1)-4)
2919 PRINT  PAPER 2; INK 7;AT 15
,6;(s(2)-4)
2920 INK 0
2930 LET casass=2: GO TO 2004
3000 REM TIRADA DEL JUGADOR
3001 IF tir<>6 THEN  LET cosiet=
0
3003 LET perst=0: LET tras=0: LE
T pos=1: LET contt=0
3005 IF tir=5 THEN  GO TO 5000
3006 IF tir=6 THEN  LET seist=se
ist+1
3007 IF seist=3 THEN  LET seist=
0: GO TO 4610
3008 FOR n=1 TO 4: IF t(n)<5 THE
N  LET contt=contt+1
3010 NEXT n
3015 IF contt=4 THEN  LET tur=1:
 GO TO 2008
3016 IF contt=0 AND tir=6 THEN
LET tir=7
3017 DIM h(4)
3020 INPUT "Que ficha mueves ?
";f$
3021 IF CODE f$<48 OR CODE f$>52
 THEN  GO TO 3020
3022 LET fic=VAL f$
3023 IF fic=0 THEN  IF tir=6 OR
tir=7 THEN  GO TO 2004
3024 IF fic=0 THEN  LET tur=1: G
O TO 2008
3025 IF t(fic)=99 THEN  PRINT #0
;"Esta ficha termino": PAUSE 25:
: GO TO 3020
3026 LET paso=0
3027 FOR n=t(fic)+1 TO t(fic)+ti
r
3028 IF c(n)=3 OR c(n)>12 THEN
LET paso=1
3029 NEXT n
3030 IF paso=1 THEN  GO TO 3600
3044 LET ant=t(fic)
3045 IF ant<5 THEN  PRINT # 0;"Es
```

Programas

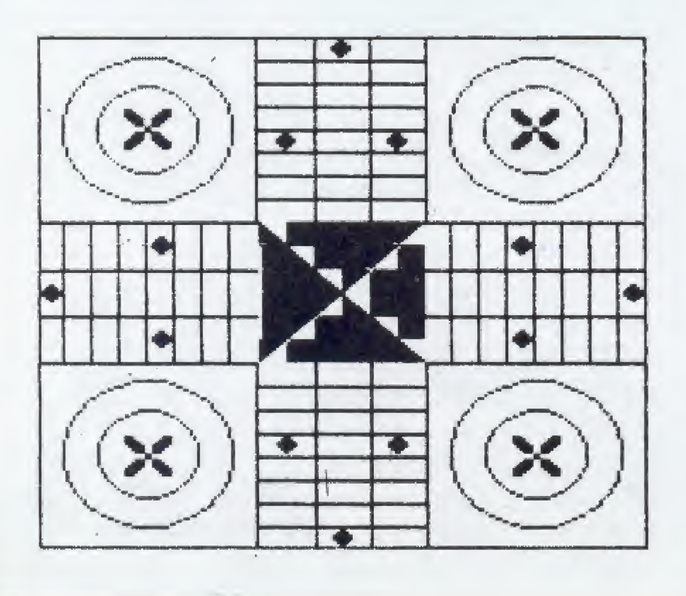
```
te en casa": PAUSE 100: GO TO 30
20
3060 IF t(fic)+tir>80 THEN  PRIN
T #0;"Te sales": PAUSE 100: GO T
O 3020
3080 LET t(fic)=t(fic)+tir
3100 IF c(ant)=11 THEN  LET pos=
1: LET c(ant)=0: GO TO 3140
3105 IF c(ant)=12 THEN  LET pos=
2: LET c(ant)=0: GO TO 3140
3110 IF c(ant)=13 THEN  LET pos=
1: LET c(ant)=12: GO TO 3140
3115 IF c(ant)=21 THEN  LET pos=
1: LET c(ant)=2: GO TO 3140
3120 IF c(ant)=22 THEN  LET pos=
2: LET c(ant)=1
3140 LET seg=0: FOR n=1 TO 12
3145 IF ant-4=z(n) THEN  LET seg
=n: LET n=12
3150 NEXT n
3151 LET pap=7: LET ppap=7
3154 IF t(fic)>72 THEN  LET pap=
4
3155 IF ant>72 THEN  LET ppap=4
3165 IF seg=0 THEN  PRINT  PAPER
 ppap;AT y(ant,pos),x(ant,pos);"
 "
3166 IF seg>0 THEN  GO TO 4000
3170 IF t(fic)=80 THEN  GO TO 89
00
3200 IF c(t(fic))=1 THEN  LET pp
=1: BEEP 1,1: GO TO 6000
3210 IF c(t(fic))=2 THEN  LET pp
=2: BEEP 1,20: GO TO 6000
3220 LET pos=1
3225 IF c(t(fic))=0 THEN  LET c(
t(fic))=11: LET pos=1: GO TO 330
0
3230 IF c(t(fic))=1 THEN  LET c(
t(fic))=22: LET pos=2: GO TO 330
0
3240 IF c(t(fic))=2 THEN  LET c(
t(fic))=21: LET pos=1: GO TO 330
0
3255 IF c(t(fic))=11 THEN  LET c
(t(fic))=13: LET pos=2: GO TO 33
00
3260 IF c(t(fic))=12 THEN  LET c
(t(fic))=13: LET pos=1: GO TO 33
00
3300 PRINT  PAPER pap; OVER 1;AT
 y(t(fic),pos),x(t(fic),pos);"@"
3301 GO SUB 600
3305 IF (t(fic)-4)<10 THEN  PRIN
T  PAPER 4; INK 0;AT 7+fic,6;"0"
;(t(fic)-4)
3310 IF (t(fic)-4)>=10 THEN  PRI
NT  PAPER 4; INK 0;AT 7+fic,6;(t
(fic)-4)
3320 IF tir=6 OR tir=7 THEN  GO
TO 2004
3350 IF tir<6 THEN  LET seist=0
3500 LET tur=1: GO TO 2005
3600 PRINT #0;"          No puedes
 pasar"
3605 PAUSE 20
3607 IF tir<6 THEN  LET seist=0
3610 LET h(fic)=1
3615 LET vuel=0
3620 FOR n=1 TO 4
3625 IF t(n)>4 AND h(n)=0 THEN
LET vuel=1: LET n=4
3630 NEXT n
3700 IF vuel=1 THEN  GO TO 3020
3705 BEEP 1,1
3707 PAUSE 100
3710 LET paso=0: LET tur=1: GO T
O 2008
4000 REM REC. DE SEGUROS
4001 IF z(seg)=5 THEN  LET pap=4
: GO TO 4010
4003 IF z(seg)=22 THEN  LET pap=
6: GO TO 4010
4005 IF z(seg)=39 THEN  LET pap=
2: GO TO 4010
4007 IF z(seg)=56 THEN  LET pap=
1: GO TO 4010
4008 LET pap=7
4010 LET cog=0: LET rec=0
4015 FOR n=1 TO 24
4020 IF z(seg)=k(n) THEN  LET co
g=n: LET n=24
4030 NEXT n
4050 IF cog<13 THEN  GO TO 4100
4060 IF cog>12 THEN  GO TO 4220
4100 FOR n=1 TO 2
4110 IF (k(cog+24))=pos THEN  LE
T rec=n
4115 LET cog=cog+6
```

```
4120 NEXT n
4130 IF rec=1 THEN  GO TO 4200
4135 IF rec=2 THEN  GO TO 4210
4200 PRINT  PAPER pap;AT y(ant,p
os),x(ant,pos);"H"
4205 LET pap=7: GO SUB 600
4206 IF perst=1 THEN  GO TO 4800
4207 IF perss=1 THEN  GO TO 4895
4208 IF tras=0 THEN  GO TO 3170
4209 IF tras=1 THEN  GO TO 7405
4210 PRINT  PAPER pap;AT y(ant,p
os),x(ant,pos);"G"
4215 LET pap=7: GO SUB 600
4216 IF perst=1 THEN  GO TO 4800
4217 IF perss=1 THEN  GO TO 4895
4218 IF tras=0 THEN  GO TO 3170
4219 IF tras=1 THEN  GO TO 7405
4220 FOR n=1 TO 2
4225 IF (k(cog+24))=pos THEN  LE
T rec=n
4230 LET cog=cog+6
4240 NEXT n
4245 IF rec=1 THEN  GO TO 4270
4250 IF rec=2 THEN  GO TO 4330
4270 PRINT  PAPER pap;AT y(ant,p
os),x(ant,pos);"E"
4295 LET pap=7: GO SUB 600
4300 IF perst=1 THEN  GO TO 4800
4301 IF perss=1 THEN  GO TO 4895
4304 IF tras=0 THEN  GO TO 3170
4305 IF tras=1 THEN  GO TO 7405
4330 PRINT  PAPER pap;AT y(ant,p
os),x(ant,pos);"F"
4495 LET pap=7: GO SUB 600
4500 IF perst=1 THEN  GO TO 4800
```




```
4501 IF perss=1 THEN  GO TO 4895
4504 IF tras=0 THEN  GO TO 3170
4505 IF tras=1 THEN  GO TO 7405
4600 REM TRIPLE SEIS
4610 LET perst=1
4615 INPUT "Que ficha pierdes ?"
;p$
4616 IF CODE p$<49 OR CODE p$>52
 THEN  GO TO 4615
4617 LET fip=VAL p$
4620 IF t(fip)<5 THEN  PRINT #0;
"Esa ficha ya esta en casa": PAU
SE 30: GO TO 4610
4625 LET ant=t(fip)
4630 LET t(fip)=fip
4640 PRINT  PAPER 4; INK 7;AT 7+
fip,6;"MN"
4650 IF c(ant)=11 THEN  LET pos=
1: LET c(ant)=0: GO TO 4750
4655 IF c(ant)=12 THEN  LET pos=
2: LET c(ant)=0: GO TO 4750
4660 IF c(ant)=13 THEN  LET pos=
1: LET c(ant)=12: GO TO 4750
4665 IF c(ant)=21 THEN  LET pos=
1: LET c(ant)=2: GO TO 4750
4670 IF c(ant)=22 THEN  LET pos=
2: LET c(ant)=1
4750 LET seg=0: FOR n=1 TO 12
4755 IF ant*4=z(n) THEN  LET seg
=n: LET n=12
4760 NEXT n
4770 IF seg>0 THEN  GO TO 4000
4790 IF seg=0 THEN  PRINT AT y(a
nt,pos),x(ant,pos);" "
4795 GO SUB 600
4797 LET casat=casat-1
4800 LET tur=1: GO TO 2005
4810 REM TRIPLE SEIS
4815 LET ans=0: LET sas=0: LET p
erss=1
4820 FOR n=4 TO 1 STEP -1
4825 IF s(n)>4 THEN  LET sas=n:
LET n=1
4830 NEXT n
4835 LET ans=s(sas)
4836 LET s(sas)=sas
4837 LET r(sas)=0
4840 IF c(ans)=1 THEN  LET pos=1
: LET c(ans)=0: GO TO 4875
4842 IF c(ans)=2 THEN  LET pos=2
```

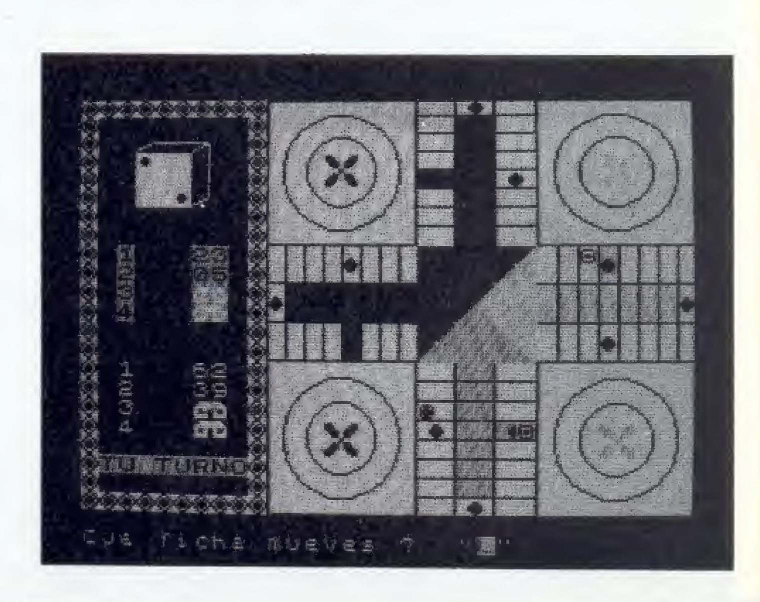
```
: LET c(ans)=0: GO TO 4875
4844 IF c(ans)=3 THEN  LET pos=1
: LET c(ans)=2: GO TO 4875
4846 IF c(ans)=21 THEN  LET pos=
2: LET c(ans)=11: GO TO 4875
4848 IF c(ans)=22 THEN  LET pos=
1: LET c(ans)=12
4875 LET seg=0: FOR n=1 TO 12
4880 IF ans-4=z(n) THEN  LET seg
=n: LET n=12
4882 NEXT n
4885 LET ant=ans
4890 IF seg>0 THEN  GO TO 4000
4892 IF seg=0 THEN  PRINT AT y(a
ns,pos),x(ans,pos);" "
4893 GO SUB 600
4895 PRINT  PAPER 2; INK 7;AT 13
+sas,6;"MN"
4896 LET casass=casass-1
4898 GO TO 2004
4999 STOP
5000 REM SALIDA DE CASA
5005 LET nt=0
5010 IF casat=4 THEN  GO TO 3020
5015 IF c(9)=13 THEN  GO TO 3020
5020 FOR s=1 TO 4
5025 IF t(s)<5 THEN  LET nt=s: L
ET s=4
5035 NEXT s
5040 IF nt<>0 THEN  LET t(nt)=9
5042 IF nt=0 THEN  GO TO 3000
5044 LET fic=nt
5050 IF c(9)=3 THEN  LET pp=1: G
O TO 6120
5052 IF c(9)=21 THEN  LET pp=2:
GO TO 6120
5054 IF c(9)=22 THEN  LET pp=1:
GO TO 6120
5060 LET casat=casat+1
5080 IF c(9)=0 THEN  LET c(9)=11
: LET pr=1: GO TO 5200
5085 IF c(9)=1 THEN  LET c(9)=22
: LET pr=2: GO TO 5200
5090 IF c(9)=2 THEN  LET c(9)=21
: LET pr=1: GO TO 5200
5145 IF c(9)=11 THEN  LET c(9)=1
3: LET pr=2: GO TO 5200
5150 IF c(9)=12 THEN  LET c(9)=1
3: LET pr=1
5200 PRINT  PAPER 4;AT y(9,pr),x
(9,pr);"Q"
5205 GO SUB 600
5210 PRINT  PAPER 4; INK 0;AT 7+
nt,6;"05"
5900 LET tur=1: GO TO 2005
6000 REM COMIDA DEL JUGADOR
6010 LET comidt=0: LET mov=1
6100 FOR N=1 TO 12
6105 IF (t(fic)-4)=z(n) THEN  LE
T comidt=1
6110 NEXT n
6115 IF comidt=1 THEN  IF pp=1 T
HEN  LET pos=2: GO TO 3225
6117 IF comidt=1 THEN  IF pp=2 T
HEN  LET pos=1: GO TO 3225
6120 FOR n=1 TO 4
6125 IF s(n)=t(fic) THEN  LET mo
v=n: LET n=4
6130 NEXT n
6135 LET s(mov)=mov
6136 LET r(mov)=0
6140 LET casass=casass-1
6150 PRINT AT y(t(fic),pp),x(t(f
ic),pp);"Q"
6160 LET c(t(fic))=pp+10
6165 PRINT  PAPER 2; INK 7;AT 13
+mov,6;"MN"
6166 GO SUB 600
6167 IF t(fic)>=14 THEN  PRINT
PAPER 4; INK 0;AT 7+fic,6;t(fic)
-4
6168 IF t(fic)<14 THEN  PRINT  P
```



```
APER 4; INK 0;AT 7+fic,6;"O";t(f
ic)-4
6170 LET tir=20: GO TO 3020
7000 REM TIRADA DEL SPECTRUM
7003 LET perss=0: LET pap=7: LET
 tras=1: LET fics=0: LET pos=1:
LET conts=0
7005 IF tir=5 THEN  GO TO 7500
7006 IF tir=6 THEN  LET seiss=se
iss+1
7007 IF seiss=3 THEN  LET seiss=
0: GO TO 4810
7008 FOR n=1 TO 4: IF s(n)<5 THE
N  LET conts=conts+1
7010 NEXT n
7015 IF conts=4 THEN  LET tur=0:
 BEEP 1,20: GO TO 2004
7030 IF conts=0 AND tir=6 THEN
LET tir=7
7100 DIM b(4): DIM o(4): DIM q(4
): DIM p(4)
7110 FOR n=1 TO 4
7120 FOR s=1 TO 12
7130 IF ((s(n)-4)+tir)=z(s) THEN
  LET o(n)=1
7140 NEXT s
7150 NEXT n
7155 IF tir<9 THEN  GO TO 7200
7160 FOR n=1 TO 4
7180 IF s(n)<39 AND (s(n)+tir)>3
8 THEN  LET trus=(s(n)+tir)-38:
LET b(n)=78+trus
7190 IF b(n)<=87 THEN  LET b(n)=
0
7195 NEXT n
7200 IF tir=6 OR tir=7 THEN  GO
TO 9200
7230 FOR w=1 TO 4
7232 FOR n=s(w)+1 TO s(w)+tir
7233 IF s(w)=72 THEN  GO TO 7236
7234 IF n>87 THEN  LET q(w)=1: G
O TO 7240
7236 IF n<87 THEN  IF c(n)=3 OR
c(n)>12 THEN  LET q(w)=1
7238 NEXT n
7240 NEXT w
7250 FOR j=1 TO 4
7255 IF b(j)=0 THEN  IF r(j)=1 A
ND q(j)=0 THEN  IF o(j)=0 THEN
IF c(s(j)+tir)=11 OR c(s(j)+tir)
=12 THEN  LET p(j)=4
7260 IF s(j)+tir=87 THEN  LET p(
j)=2
7265 NEXT j
7280 FOR n=1 TO 4
7282 IF p(n)=4 THEN  LET fics=n:
 LET n=4
7284 NEXT n
7286 IF fics>0 THEN  GO TO 7320
7290 FOR n=1 TO 4
7292 IF p(n)=2 THEN  LET fics=n:
 LET n=4
7294 NEXT n
7296 IF fics>0 THEN  GO TO 7320
7300 FOR n=1 TO 4
7301 IF tir=20 THEN  IF b(n)=0 T
HEN  IF r(n)=1 AND q(n)=0 THEN
IF s(n)=72 THEN  LET fics=n: LET
 n=4: GO TO 7304
7302 IF b(n)=0 THEN  IF s(n)<>99
 THEN  IF r(n)=1 AND q(n)=0 THEN
  IF s(n)+tir<87 THEN  LET fics=
n: LET n=4
7304 NEXT n
7314 IF fics=0 THEN  IF tir=6 OR
 tir=7 THEN  LET tur=1: GO TO 20
08
7315 IF fics=0 THEN  GO TO 2004
7320 LET ans=s(fics)
7330 LET s(fics)=s(fics)+tir
7335 LET trus=0
7337 IF ans>79 THEN  GO TO 7360
7340 IF ans<39 AND s(fics)>38 TH
EN  LET trus=s(fics)-38: LET s(f
ics)=79+trus
7352 IF s(fics)>72 AND trus=0 TH
EN  LET s(fics)=s(fics)-68
7360 IF c(ans)=1 THEN  LET pos=1
: LET c(ans)=0: GO TO 7370
7362 IF c(ans)=2 THEN  LET pos=2
: LET c(ans)=0: GO TO 7370
7364 IF c(ans)=3 THEN  LET pos=1
: LET c(ans)=2: GO TO 7370
7366 IF c(ans)=21 THEN  LET pos=
2: LET c(ans)=11: GO TO 7370
7368 IF c(ans)=22 THEN  LET pos=
1: LET c(ans)=12
7370 LET seg=0: FOR n=1 TO 12
7375 IF ans-4=z(n) THEN  LET seg
```

Programas

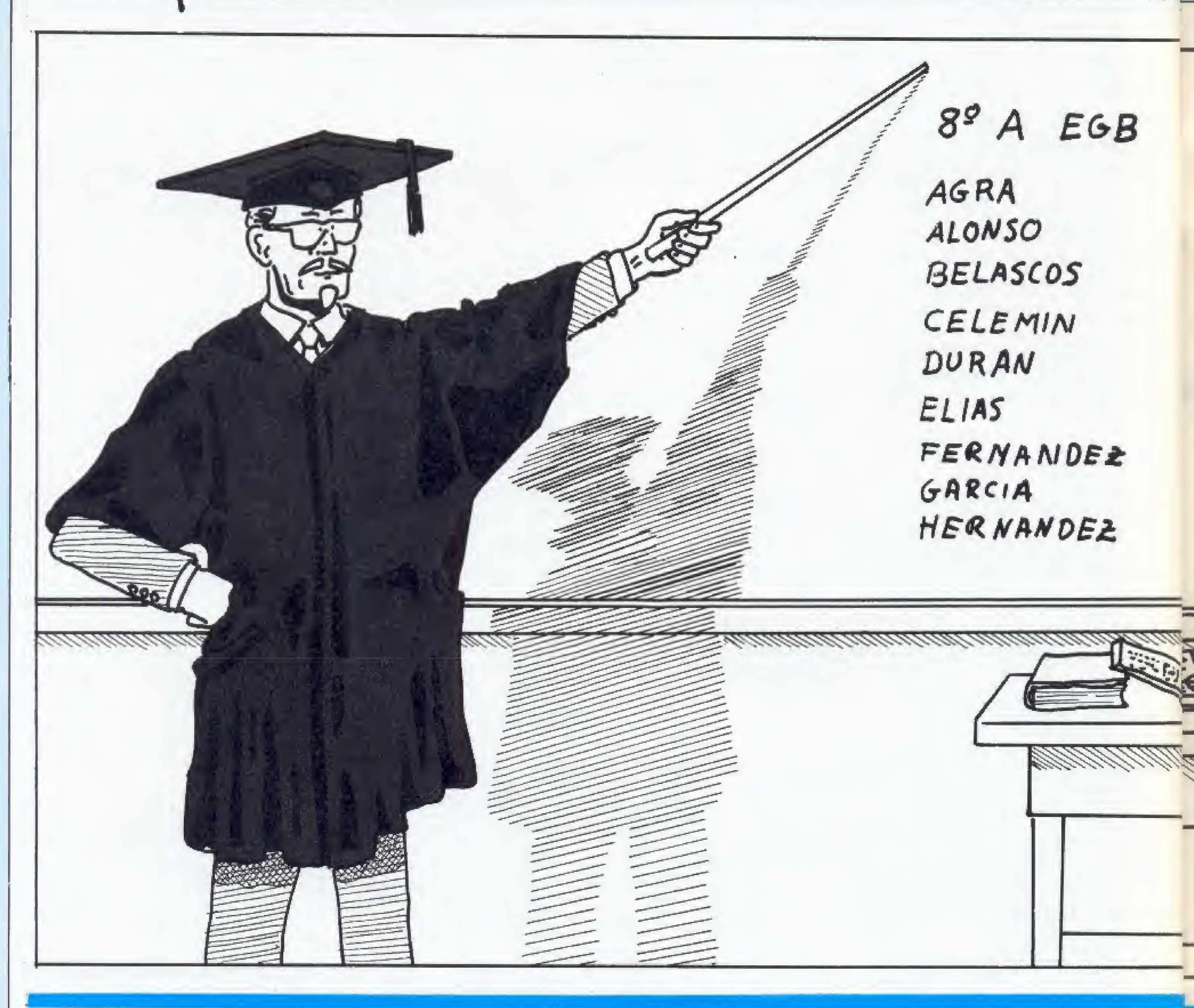
```
=n: LET n=12
7380 NEXT n
7381 LET ppap=7
7385 IF s(fics)>79 THEN  LET pap
=2
7387 IF ans>79 THEN  LET ppap=2
7390 IF seg=0 THEN  PRINT  PAPER
 ppap;AT y(ans,pos),x(ans,pos);"
 "
7400 IF seg>0 THEN  LET ant=ans:
 GO TO 4000
7405 IF s(fics)=87 THEN  GO TO 8
800
7420 IF c(s(fics))=11 AND o(fics
)=0 THEN  LET ps=1: GO TO 8500
7425 IF c(s(fics))=12 AND o(fics
)=0 THEN  LET ps=2: GO TO 8500
7430 IF c(s(fics))=0 THEN  LET c
(s(fics))=1: LET pos=1: GO TO 74
80
7435 IF c(s(fics))=1 THEN  LET c
(s(fics))=3: LET pos=2: GO TO 74
80
7440 IF c(s(fics))=2 THEN  LET c
(s(fics))=3: LET pos=1: GO TO 74
80
7450 IF c(s(fics))=11 AND o(fics
)=1 THEN  LET pos=2: LET c(s(fic
s))=21: GO TO 7480
7460 IF c(s(fics))=12 AND o(fics
)=1 THEN  LET pos=1: LET c(s(fic
s))=22: GO TO 7480
7480 PRINT  PAPER pap;AT y(s(fic
s),pos),x(s(fics),pos);"R"
7481 GO SUB 600
7482 IF (s(fics)-4)<10 THEN  PRI
NT  PAPER 2; INK 7;AT 13+fics,6;
"0";(s(fics)-4)
7485 IF (s(fics)-4)>=10 THEN  PR
INT  PAPER 2; INK 7;AT 13+fics,6
;(s(fics)-4)
7487 INK 0
7488 IF tir=6 OR tir=7 THEN  LET
 tur=1: GO TO 2008
7489 IF tir<6 THEN  LET seiss=0
7490 GO TO 2004
7500 REM SALIDA CASA DE SP
7505 LET ns=0
7510 IF casass=4 THEN  GO TO 710
0
7520 IF c(43)=3 THEN  GO TO 7100
7560 FOR n=1 TO 4
7570 IF s(n)<5 THEN  LET ns=n: L
ET n=4
7580 NEXT n
7590 IF ns<>0 THEN  LET s(ns)=43
: LET r(ns)=1
7595 LET fics=ns
7600 IF c(43)=13 THEN  LET ps=1:
 GO TO 8605
7605 IF c(43)=21 THEN  LET ps=1:
 GO TO 8605
7610 IF c(43)=22 THEN  LET ps=2:
 GO TO 8605
7640 LET casass=casass+1
7700 IF c(43)=0 THEN  LET c(43)=
1: LET pr=1: GO TO 7900
7702 IF c(43)=1 THEN  LET c(43)=
3: LET pr=2: GO TO 7900
7704 IF c(43)=2 THEN  LET c(43)=
3: LET pr=1: GO TO 7900
7706 IF c(43)=11 THEN  LET c(43)
=21: LET pr=2: GO TO 7900
7708 IF c(43)=12 THEN  LET c(43)
=22: LET pr=1: GO TO 7900
7900 PRINT  PAPER 2; INK 0;AT y(
43,pr),x(43,pr);"R"
7950 PRINT  PAPER 2; INK 7;AT 13
+ns,6;"39": INK 0
7999 GO TO 2004
8500 REM COMIDA DEL SPECTRUM
8510 LET comids=0: LET mos=1
8520 FOR N=1 TO 12
```



(53)

```
8540 IF (s(fics)-4)=z(n) THEN  L
ET comids=1
8550 NEXT N
8570 IF comids=1 THEN  LET o(fic
s)=1: GO TO 7450
8605 FOR n=1 TO 4
8610 IF t(n)=s(fics) THEN  LET m
os=n: LET n=4
8615 NEXT n
8625 LET t(mos)=mos
8630 LET casat=casat-1
8635 PRINT  PAPER 2; INK 7;AT 13
+fics,6;s(fics)-4
8640 PRINT  PAPER 4; INK 7;AT 7+
mos,6;"MN"
8650 PRINT AT y(s(fics),ps),x(s(
fics),ps);"R"
8655 GO SUB 600
8660 LET c(s(fics))=ps-10
8700 LET tir=20: GO TO 7000
8800 REM FINAL FICHA
8810 LET s(fics)=99
8830 PRINT  PAPER 2; INK 7;AT 13
+fics,6;"OP"
8870 LET gans=gans+1
8880 GO SUB 600
8885 IF gans=4 THEN  GO TO 9500
8890 LET tir=10: GO TO 7000
8900 LET t(fic)=99
8910 PRINT  PAPER 4; INK 0;AT 7+
fic,6;"OP"
8920 LET gant=gant+1
8930 GO SUB 600
8940 IF gant=4 THEN  GO TO 9500
8950 LET tir=10: GO TO 3020
8999 STOP
9000 REM DADO
9010 PRINT AT 3,3;"   ";AT 4,3;"
 C ";AT 5,3;"   "
9015 RETURN
9020 PRINT AT 3,3;"C  ";AT 4,3;"
   ";AT 5,3;"  C"
9025 RETURN
9030 PRINT AT 3,3;"C  ";AT 4,3;"
 C ";AT 5,3;"  C"
9035 RETURN
9040 PRINT AT 3,3;"C C";AT 4,3;"
   ";AT 5,3;"C C"
9045 RETURN
9050 PRINT AT 3,3;"C C";AT 4,3;"
 C ";AT 5,3;"C C"
9055 RETURN
9060 PRINT AT 3,3;"C C";AT 4,3;"
C C";AT 5,3;"C C"
9065 RETURN
9200 LET puens=0
9205 FOR n=1 TO 4
9210 IF c(s(n))=3 THEN  IF s(n)+
tir<88 THEN  IF q(n)=0 THEN  LET
 puens=n: LET n=4
9220 NEXT n
9230 IF puens<>0 THEN  LET fics=
puens: GO TO 7320
9240 IF puens=0 THEN  GO TO 7230
9500 REM FINAL
9505 IF gant=4 THEN  PRINT  FLAS
H 1;AT 8,18;"      ";AT 9,18;"
ME   ";AT 10,18;"      ";AT 11,18
;"  HAS ";AT 12,18;"      ";AT 1
3,18;"GANADO"
9506 IF gans=4 THEN  PRINT  FLAS
H 1;AT 8,18;"      ";AT 9,18;"
TE   ";AT 10,18;"      ";AT 11,18
;"  HE  ";AT 12,18;"      ";AT 1
3,18;"GANADO"
9508 FOR n=1 TO 13: FOR d=40 TO
50: BEEP .01,d: NEXT d: NEXT n
9510 FOR n=5 TO 1 STEP -1
9515 FOR s=0 TO 21
9525 BEEP .01,s+30: PRINT  INK n
; OVER 1;AT s,0;"[redacted]"
9535 NEXT s
9600 NEXT n
9610 PAUSE 30
9700 FOR n=1 TO 22: BEEP .01,n+2
0: PAUSE 10: RANDOMIZE USR 3190:
 NEXT n
9710 CLS
9750 FOR n=1 TO 22: BEEP .02,n+3
0: PAUSE 10: RANDOMIZE USR 3190:
 NEXT n
9770 PRINT  FLASH 1;AT 10,0;"Qui
eres jugar otra partida  s/n "
9780 INPUT ;y$
9785 IF y$="s" OR y$="S" THEN  C
LS : GO TO 150
9800 STOP
```

Programas




```
  10 REM ***********************
*          *
*          *CREACION DE LISTAS DE
*          *  ALUMNOS ORDENADAS
*          *   ALFABETICAMENTE
*          *
*           Jaime Melendi 1984
*          *
*          ***********************
*
  50 DIM q$(7,1)
  60 DIM w$(1,25)
  70 DIM a$(900,30)
  80 DIM b$(100,30)
  90 LET x=0
 100 GO SUB 7000
 101
 102 REM **********************
        ******* INDICE *******
        **********************
 103
 110 CLS : POKE 23609,100: POKE
23658,8 .
 120 PRINT  '''                INDI
CE                             ====
==

                    Creacion del fi
chero ..... 1
                    Listado de un c
urso ...... 2
                    Listado total .
........... 3
```

Listas escolares

El mes de septiembre ha sido el mes de la vuelta al colegio. Padres y chicos hacen cola en las librería, comprando el material; luego las matrículas... A continuación los profesores tienen que realizar un trabajo muy pesado. El programa que os presentamos ayuda a preparar e imprimir las listas escolares.

Una de las labores más tediosas en la puesta en marcha de un año académico es, la confección de las listas de alumnos de cada curso, teniendo en cuenta que debemos ordenarlas alfabéticamente para después proceder a separar los distintos grupos.

Hay que trabajar con una gran cantidad de información y realizar las oportunas manipulaciones para, al final, obtener el resultado apetecido.

Pero el problema se agrava si cuando tenemos las listas preparadas es necesario hacer alguna corrección o incluir algún alumno que se haya despistado.

Teniendo en cuenta que las características fundamentales de los ordenadores son la velocidad y la fiabilidad para trabajar con grandes volúmenes de información, encarguemos a nuestro modesto Spectrum esta tarea.

Este programa ha sido pensado para la confección de listas de cursos, ordenadas alfabéticamente, en un colegio de E.G.B., pudiendo adaptarse fácilmente a otro tipo de enseñanzas.

Los registros estarán formados por los siguientes campos:

1.er campo: Curso; tamaño: 1 caracter.

2.º campo: Apellidos y nombre; tamaño: 25 caracteres.

3.er campo: Idioma moderno; tamaño: 1 caracter.

4.º campo: Religión o ética; tamaño: 1 caracter.

5.º campo: Curso anterior suspenso o aprobado; tamaño: 1 caracter.

6.º campo: Repite sí/no; tamaño: 1 caracter.

Longitud del registro: 30 caracteres.

El número total de alumnos que admite el programa, dada la capacidad de memoria interna del Spectrum, es de 900. El número máximo de alumnos por curso es de 100, lo que hace posible la introducción de 9 cursos de un máximo de 100 alumnos por curso.

La longitud del registro está condicionada por la capacidad de impresión de la pantalla y de la impresora ZX o SEIKOSA GP-505, es decir, 32 caracteres por línea.

Para poner en marcha el programa procederemos en primer lugar a teclearlo. Finalizada esta ardua labor haremos una copia del mismo en cinta de cassette. A continuación procederemos a ejecutarlo, corrigiendo los posibles errores

```
                   Correcciones ..
........... 4
                   Fin y Grabacion
 .......... 5"
 130 FAUSE 0
 140 LET go=VAL INKEY$
 150 IF go<1 AND go>5 THEN  GO T
O 130
 160 GO TO 1000*go
 999
1000 REM ***********************
         *** CREACION FICHERO **
         ***********************
1001
1010 CLS : FRINT AT 0,5;"CREACIO
N DEL FICHERO"
1020 PRINT AT 2,3;"PARA SALIR RE
PITE NUMERO"
1030 GO SUB 7200
1040 LET k$="
        "
1050 LET q=0: LET w=0: LET e=0:
LET u=0: LET t=0: LET y=0: LET a
1=0
1060 PRINT AT 7,14;"    ";AT 7,2
8;"   ";AT 11,3;k$;AT 13,11;"  "
;AT 13,26;"  ";AT 15,11;"  ";AT
15,26;"  "
1070 PRINT  BRIGHT 1;AT 4,5;" Ul
timo numero ";x;" "
1080 LET r=x
1090 INPUT "N. alumno ";x
1095 IF x<=r-1 THEN  GO TO 1090
1100 IF x=r THEN  GO TO 110
1110 PRINT AT 7,14;x: IF al=1 TH
EN  GO TO 1230
```

que hayamos cometido. Una vez depurado lo grabaremos de nuevo, teniendo de este modo una copia del programa original, al que poder recurrir en caso de necesidad. Para grabarlo es necesario hacerlo con «autorun» en la línea 100, es decir, SAVE «nombre» LINE100.

El procedimiento que utiliza el programa para almacenar los datos es el de introducirlos en variables subindicadas, vectores o listas, a$(900,30) para el archivo general y b$(100,30) para el archivo del curso. Teniendo en cuenta que el comando RUN borra los valores de las variables no puede utilizarse para ejecutar el programa cuando tenga datos, pues no los borraría. En caso de necesidad hay que utilizar el comando GOTO 100; solamente ejecutaremos el programa con RUN para inicializarlo y comenzar a grabar los datos, es decir, la primera vez que lo empleemos.

El trabajo de introducir y modificar las listas de alumnos, una tarea repetitiva

Ya está tecleado el programa. Lo hemos almacenado en cinta y queremos comenzar a utilizarlo. Pulsamos RUN y elegimos la opción 1 «Creación del fichero». Termina-

```
1120 INPUT "Curso ";q$(1): PRINT
 AT 7,28;q$(1): IF q=1 THEN  GO
TO 1230
1130 POKE 23658,0: INPUT "Apelli
dos y nombre, maximo 25 caracter
es ";w$(1): POKE 23658,8
1140 PRINT AT 11,3;w$(1): IF w=1
 THEN  GO TO 1230
1145 IF VAL q$(1)<6 THEN  LET q$
(2)="*": GO TO 1160
1150 INPUT "Ingles (I), Frances
(F) ";q$(2): IF q$(2)<>"I" AND q
$(2)<>"F" THEN  BEEP .5,-12: GO
TO 1150
1160 PRINT AT 13,11;q$(2): IF e=
1 THEN  GO TO 1230
1170 INPUT "Religion (R), Etica
(E) ";q$(3): IF q$(3)<>"R" AND q
$(3)<>"E" THEN  BEEP .5,-12: GO
TO 1170
1180 PRINT AT 13,26;q$(3): IF u=
1 THEN  GO TO 1230
1185 IF q$(1)="1" THEN  LET q$(4
)="*": GO TO 1200
1190 INPUT "Curso anterior: Apro
bado (A), Suspenso (S) ";q$(4):
IF q$(4)<>"A" AND q$(4)<>"S" THE
N  BEEP .5,-12: GO TO 1190
1200 PRINT AT 15,11;q$(4): IF t=
1 THEN  GO TO 1230
1205 IF q$(1)="1" OR q$(1)="3" O
R q$(1)="4" THEN  LET q$(5)="*":
 GO TO 1220
1210 INPUT "Repite el curso S/N:
 ";q$(5): IF q$(5)<>"S" AND q$(5
)<>"N" THEN  BEEP .5,-12: GO TO
1210
1220 PRINT AT 15,26;q$(5): IF y=
1 THEN  GO TO 1230
1230 INPUT "TODO CORRECTO? (S/N)
 ";q$(6): IF q$(6)="S" THEN  GO
TO 1340
1240 IF q$(6)="N" THEN  INPUT "D
ame el n. de casilla incorrecta
";inc
1250 IF q$(6)<>"S" AND q$(6)<>"N
" THEN  BEEP .5,-12: GO TO 1230
1260 IF inc=1 THEN  LET al=1: GO
 TO 1090
1270 IF inc=2 THEN  LET q=1: GO
TO 1120
1280 IF inc=3 THEN  LET w=1: GO
TO 1130
1290 IF inc=4 THEN  LET e=1: GO
TO 1150
1300 IF inc=5 THEN  LET u=1: GO
TO 1170
1310 IF inc=6 THEN  LET t=1: GO
TO 1190
1320 IF inc=7 THEN  LET y=1: GO
TO 1210
1330 BEEP .5,-12: GO TO 1240
1340 IF q$(5)="S" THEN  LET q$(5
)="r"
1350 IF q$(5)="N" THEN  LET q$(5
)=" "
1360 LET a$(x)=q$(1)+w$(1)+q$(2)
+q$(3)+q$(4)+q$(5)
1370 GO TO 1050
1999
2000 REM ***********************
         **** LISTADO CURSO ****
         ***********************
2001
2010 BEEP 1,12: CLS : PRINT  BRI
GHT 1;AT 7,5;"
```



El programa nos pide el Idioma Moderno sólo si el alumno estudia segunda etapa de EGB

do el trabajo de introducción de datos, por este día, vamos a la opción 5 «Fin y Grabación»; en esta opción procederá a grabar en cinta de cassette el programa con los datos introducidos y, posteriormente, a verificar la correcta grabación, dándonos en todo momento las instrucciones necesarias. Nos preguntará si todo ha sido correcto, contestando «no» repetirá el proceso haciendo una segunda grabación como copia de seguridad, contestando «sí» el programa finaliza.

Para continuar cargando datos al programa es necesario leer el último programa grabado, dado que tiene los datos almacenados anteriormente.

Creo que es importante advertir que el programa tarda más de tres minutos en grabarse, por lo que es necesario tener un poco de paciencia. En el caso de disponer de Microdrive es fácil adaptar el programa a este periférico, consiguiendo una mayor velocidad y facilidad de grabación.

Además de las opciones ya vis-

```
  ";AT 8,5;" LISTADO DE UN CURS
O ";AT 9,5;"
 ": INPUT "codigo del curso ";q$
(7)
2020 INPUT "nombre del curso ";d
$
2030 LET q=1
2040 FOR n=1 TO x
2050 IF a$(n)( TO 1)=q$(7) THEN
 GO SUB 2280
2060 NEXT n
2070 FOR s=1 TO q-1
2080 FOR m=s TO 2 STEP -1
2090 IF b$(m)>b$(m-1) THEN  LET
m=2: GO TO 2110
2100 LET j$=b$(m): LET b$(m)=b$(
m-1): LET b$(m-1)=j$
2110 NEXT m: NEXT s
2119
2120 REM **********************
         ****** IMPRESION ******
         **********************
2121
2130 CLS
2140 LET d=(32-LEN d$)/2
2150 INPUT "Impresora S/N ";q$(6
): IF q$(6)<>"S" AND q$(6)<>"N"
THEN  GO TO 2150
2160 IF q$(6)="S" THEN  OPEN #2,
"p"
2170 PRINT TAB d;d$
2180 PRINT '''
2190 FOR l=1 TO q-1
2200 PRINT : IF l<10 THEN  PRINT
 " ";
2210 PRINT l;" ";b$(l)(2 TO )
2220 NEXT l
2230 CLOSE #2
2240 PRINT ''';"Pulse una tecla
para continuar"
2250 FOR n=1 TO q: LET b$(n)="":
 NEXT n
2260 PAUSE 0
2270 GO TO 110
2279
2280 REM **********************
         **** MATRIZ CURSO *****
         **********************
2281
2290 LET b$(q)=a$(n)
2300 LET q=q+1
2310 RETURN
2999
3000 REM **********************
         **** LISTADO TOTAL ****
         **********************
3001
3010 CLS : PRINT  BRIGHT 1;AT 7,
8;"              ";AT 8,8;" LIS
```

```
INDICE
======

Creacion del fichero ..... 1

Listado de un curso ...... 2

Listado total ............ 3

Correcciones ............. 4

Fin y Grabacion .......... 5
```

tas, el programa dispone de otras tres: «Listado de un curso», que permite listar alfabéticamente los alumnos de un curso con sus datos; «Listado total», lista de los datos indicados en el mismo orden de entrada; y «Correcciones», que permite modificar los errores que hayamos cometido al introducir los datos.

Comentaremos detenidamente la opción «Creación del fichero»: en la última línea de la pantalla, al objeto de poder comprobar que sean correctos. El número de alumno será el de matrícula, correlativo desde el 1 al 900. Para saber cual ha sido el último número utilizado, y no repetirlo, el programa nos lo indica. Basándonos en la imposibilidad de duplicación del número de alumno, saldremos de esta opción repitiendo el último empleado. El idioma moderno solamente nos lo preguntará en los cursos del ciclo superior. Solamente nos preguntará si es repetidor en los cursos en que puede suceder.

Puede manejar hasta 900 alumnos, con un máximo de 100 por curso

Una vez introducidos los datos de un alumno, nos preguntará si

```
TADO TOTAL ";AT 9,8;"
   "
3015 INPUT "Todos los datos ""T"
", parte de    ellos ""P"" ";q$(6
): IF q$(6)<>"T" AND q$(6)<>"P"
THEN  GO TO 3015
3017 LET b1=1: LET b2=x: IF q$(6
)="P" THEN  INPUT "Desde el nume
ro ";b1: INPUT "Al numero ";b2:
IF b2>x THEN  LET b2=x
3020 INPUT "Impresora S/N ";q$(6
): IF q$(6)<>"S" AND q$(6)<>"N"
THEN  GO TO 3020
3030 IF q$(6)="S" THEN  OPEN #2,
"P"
3040 CLS : PRINT TAB 8;"LISTADO
TOTAL"
3050 PRINT ''
3060 FOR n=b1 TO b2
```

```
            ATENCION


En caso de que este programa se
 ROMPA en cualquier momento y
 quiera continuar ejecutandolo

nunca pulse RUN  pulse GOTO 100



 PULSE UNA TECLA PARA CONTINUAR
```

```
3070 PRINT n;" ";a$(n)
3080 NEXT n
3090 CLOSE #2
3100 PRINT '''; " Pulse una tecl
a para continuar": PAUSE 0
3110 GO TO 110
3999
4000 REM **********************
       **** CORRECCIONES *****
       **********************
4001
4210 CLS : PRINT AT 1,8;"▛▀▀▀▀▀▀▀
▀▀▀▀▀▀▜"
4220 PRINT AT 2,8;"▌CORRECCIONES
▐"
4230 PRINT AT 3,8;"▙▄▄▄▄▄▄▄▄
▄▟"
4240 GO SUB 7200
4250 INPUT "N. alumno? ";c
4260 LET q$(1)=a$(c)( TO 1): LET
 w$(1)=a$(c)(2 TO 27): LET q$(2)
=a$(c)(27 TO 27): LET q$(3)=a$(c
)(28 TO 28): LET q$(4)=a$(c)(29
TO 29): LET q$(5)=a$(c)(30 TO 30
): IF q$(5)=" " THEN  LET q$(5)=
"N"
4265 IF q$(5)="r" THEN  LET q$(5
)="S"
4270 PRINT AT 7,14;c;AT 7,28;q$(
1);AT 11,3;w$(1);AT 13,11;q$(2);
AT 13,26;q$(3);AT 15,11;q$(4);AT
 15,26;q$(5)
4280 INPUT "TODO CORRECTO? (S/N)
 ";q$(6): IF q$(6)="S" THEN  GO
TO 4390
4290 IF q$(6)<>"S" AND q$(6)<>"N
" THEN  BEEP .5,-12: GO TO 4280
4300 INPUT "Dame el n. de casill
```



son todos correctos y, en caso de tener algún error e indicarle que no, nos reclamará el número de casilla incorrecta, para proceder a efectuar la oportuna corrección.

Por último, nos reclama el número de alumno, repitiéndose el proceso. Para salir de esta opción, repetiremos el último y el programa vuelve al INDICE.

En la opción «Listado de un curso», nos pedirá el código y el nombre del curso. Nos preguntará si

Se puede imprimir una lista provisional, corregir los errores y sacar listas ordenadas inmediatamente

queremos que el listado salga por pantalla o por impresora.

Toma del fichero principal los datos de los alumnos del curso indicado, los introduce en el fichero de curso, los ordena alfabéticamente y los imprime. Habida cuenta que la ordenación la realiza antes de la impresión de los datos, es posible el incluir un alumno en el último momento en el fichero principal y obtener inmediatamente la lista perfectamente ordenada incluido el nuevo alumno. Finalizado el proceso retorna al INDICE.

En la opción «Listado total»,

```
a incorrecta ";inc
4310 IF inc=1 THEN  PRINT  BRIGH
T 1;AT 19,1;" El numero no puede
 cambiarse ": BEEP 2,24: PRINT A
T 19,1;"
       ": GO TO 4280
4320 IF inc=2 THEN  INPUT "Curso
 ";q$(1): PRINT AT 7,28;q$(1): G
O TO 4280
4330 IF inc=3 THEN  POKE 23658,0
: INPUT "Apellidos y nombre, max
imo 25 caracteres";w$(1): PRINT
AT 11,3;w$(1): POKE 23658,8
4340 IF inc=4 THEN  INPUT "Ingle
s (I), Frances (F) ";q$(2): PRIN
T AT 13,11;q$(2): IF q$(2)="I" O
R q$(2)="F" THEN  GO TO 4280
4345 IF q$(2)<>"F" AND q$(2)<>"I
" AND q$(2)<>"*" THEN  BEEP .5,1
2: GO TO 4340
4350 IF inc=5 THEN  INPUT "Relig
ion (R), Etica (E) ";q$(3): PRIN
T AT 13,26;q$(3): IF q$(3)="R" O
R q$(3)="E" THEN  GO TO 4280
4355 IF q$(3)<>"R" AND q$(3)<>"E
" THEN  BEEP .5,12: GO TO 4350
4360 IF inc=6 THEN  INPUT "Curso
 anterior: Aprobado (A), Suspens
o (S) ";q$(4): PRINT AT 15,11;q$
(4): IF q$(4)="A" OR q$(4)="S" T
HEN  GO TO 4280
4365 IF q$(4)<>"A" AND q$(4)<>"S
" AND q$(4)<>"*" THEN  BEEP .5,1
2: GO TO 4360
4370 IF inc=7 THEN  INPUT "Repit
e el curso S/N ";q$(5): PRINT AT
 15,26;q$(5): IF q$(5)="S" OR q$
(5)="N" THEN  GO TO 4280
4375 IF q$(5)<>"S" AND q$(5)<>"N
" AND q$(5)<>"*" THEN  BEEP .5,1
2: GO TO 4370
4380 BEEP .5,12: GO TO 4280
4390 IF q$(5)="S" THEN  LET q$(5
)="r"
4400 IF q$(5)="N" THEN  LET q$(5
)=" "
4410 LET a$(c)=q$(1)+w$(1)+q$(2)
+q$(3)+q$(4)+q$(5)
4420 INPUT "MAS CORRECCIONES? S/
N ";q$(6): IF q$(6)<>"S" AND q$(
6)<>"N" THEN  GO TO 4420
4430 IF q$(6)="N" THEN  GO TO 11
0
4440 PRINT AT 7,14;"    ";AT 7,2
8;"  ";AT 11,3;k$;AT 13,11;"  "
;AT 13,26;"  ";AT 15,11;"  ";AT
15,26;"  "
4450 GO TO 4250
4999

5000 REM ***********************
         ****** GRABACION ******
         ***********************

5001

5010 LET p$="
  "

5020 CLS : PRINT  BRIGHT 1;AT 5,
2;" desconecte la clavija EAR ";
AT 9,5;p$;AT 10,5;" PREPARE EL C
ASSETTE ";AT 11,5;p$;AT 12,5;"
   PARA GRABAR      ";AT 13,5;p$
5030 SAVE "LISTAS" LINE 100
5040 PRINT AT 18,7;" PARE EL CAS
SETTE ": BEEP 2,24: PAUSE 200
5050 CLS : PRINT  BRIGHT 1;AT 5,
3;" conecte la clavija EAR ";AT
9,5;p$;AT 10,5;"  REBOBINE LA CI
```

nos preguntará si queremos todos los datos o parte de ellos. Si hemos optado por un listado parcial nos preguntará el primer número a listar y el último. Como en la opción anterior, es posible hacer un listado por pantalla o por impresora. Finalizado el proceso regresa al INDICE.

Opción «Correcciones», este proceso es similar al de «Creación del fichero». Preguntará el número de alumno y escribirá todos sus datos en las casillas correspondientes. Nos preguntará si todos los datos son correctos, indicándole que no nos pedirá el número de casilla incorrecta y haremos las correcciones oportunas. Preguntará, por último, si queremos hacer más correcciones o no. Contestando que no,retornará al INDICE.

La carga del programa con los datos tarda unos tres minutos

Jaime Melendi **48K**

```
NTA   ";AT 11,5;p$;AT 12,5;"     Y
 PULSE PLAY       ";AT 13,5;p$;AT
14,5;"   PARA VERIFICAR     ";AT
15,5;p$
5060 VERIFY "LISTAS"
5070 CLS : PRINT AT 18,7;" PARE
EL CASSETTE ": BEEP 2,24
5080 INPUT "TODO CORRECTO? S/N "
;f$: IF f$<>"S" AND f$<>"N" THEN
  GO TO 5080
5090 IF f$="S" THEN  CLS : POKE
23658,0: GO TO 9999
5100 GO TO 5000
6999
7000 REM ***********************
         *** RUTINA DE     ******
         *** PRESENTACION ******
         ***********************
7001
7010 CLS : PRINT AT 3,10; FLASH
1;" ATENCION "
7020 BEEP 2,12
7030 PRINT ''';"En caso de que e
ste programa se  ROMPA en cualqu
ier momento y    quiera continua
r ejecutandolo"
7040 PRINT ' INVERSE 1;"nunca pu
lse RUN"; INVERSE 0;" "; BRIGHT
1;" pulse GOTO 100 "
7050 PRINT ''''';" PULSE UNA TEC
LA PARA CONTINUAR"
7060 PAUSE 0
7070 RETURN
7199
7200 REM ***********************
         ****** RUTINA DE ******
         ****** PANTALLA   ******
         ***********************
7201
7210 PLOT 3,123: DRAW 248,0: DRA
W 0,-80: DRAW -248,0: DRAW 0,80:
 PLOT 3,107: DRAW 248,0: PLOT 3,
75: DRAW 248,0
7220 PLOT 3,59: DRAW 248,0: PLOT
 123,75: DRAW 0,-32: PLOT 147,12
3: DRAW 0,-16
7230 PRINT  BRIGHT 1;AT 7,1;"1";
AT 7,19;"2";AT 9,1;"3";AT 13,1;"
4";AT 13,16;"5";AT 15,1;"6";AT 1
5,16;"7"
7240 PRINT AT 7,3;"N. alumno:";A
T 7,21;"Curso:";AT 9,3;"Apellido
s y nombre:";AT 13,3;"Idioma:";A
T 13,18;"Rel/Et:";AT 15,3;"Ap/Su
s:";AT 15,18;"Repite:"
7250 RETURN
```

```
CORRECCIONES

1 N. alumno: 2      2 Curso: 8

3 Apellidos y nombre:

  Martinez Sanjuan, Luis

4 Idioma: F      5 Rel/Et: R

6 Ap/Sus: S      7 Repite: N
```

Preguntas y respuestas

P Poseo dos *interfaces,* uno para impresora Centronics y otro para *joystick.* Muy a menudo los utilizo, o sea que tengo que estar desconectando uno y conectando el que me interesa con bastante frecuencia. Mi pregunta es si tal operación, con el tiempo, puede perjudicar al conector de expansiones del Spectrum o a los *interfaces.* Y si es así, qué solución me podéis dar.

Iván Sansa
Barcelona

R **Tu problema no sería tal si todas las casas dedicadas a la fabricación de estos productos se dieran cuenta de que puede interesarnos utilizar varios** interfaces **simultáneamente o, como es tu caso, tener ambos conectados constantemente. De hecho muchos** interfaces **preveen esta posibilidad y disponen de otra salida por la que podremos conectar nuevos periféricos.**

De todas formas en tiendas especializadas podrás encontrar un «ladrón de expansiones»: el Currah microSlot, gracias al cual podrás «multiplicar» las salidas del bus trasero para que admita más periféricos.

P Según el manual del Spectrum la instrucción BASIC IN x carga el par de registros bc con x y realiza la instrucción assembler IN a,(c). ¿Qué valor tendría c en IN 31? ¿Y en IN 32762?

Jesús Manzanal
León

R **Para introducir este tipo de parámetros en memoria el ordenador los descompone en dos** bytes **(dos números entre 0 y 255) que son los que se almacenan (en nuestro caso en el par bc). Para «descomponer» así un número entero, que deberá estar comprendido entre 0 y 65535, procederemos de la siguiente forma: El** byte **más significativo lo conseguiremos hallando la parte entera de la división del valor a calcular entre 255 y el menos significativo sustrayendo al valor a calcular el conseguido anteriormente multiplicado por 255. Si descomponemos de esta manera el número 32 quedará 0 en el registro b y 32 en el c. Si hacemos lo mismo con 32762 el resultado será de 127 para el registro b y 250 para el c. Para calcular el valor de un par de registros de dos celdas de memoria deberemos multiplicar el** byte **más significativo por 255 y sumarle el menos significativo.**

P Tengo el ensamblador de código máquina «Astron» de Dk'tronics. Cuando lo cargué en mi Spectrum de 48 K e introduje el programa del número de mayo, «Spectrum parlanchín», apareció en la pantalla un mensaje en el que me impedía la colocación de etiquetas iguales en líneas distintas (líneas del programa: 17, 18, 19, 20 y 21). ¿Cómo se puede solucionar esto?

Eduardo Fernández
Las Palmas de Gran Canaria

R **Las «etiquetas» a que te refieres no son tales sino «comandos»** *assembler*, **por lo que deberás dejar un espacio antes de teclearlos para indicarle al ensamblador que no deseas poner ninguna etiqueta en esa línea. Seguro que estarás pensando que en las listas de mnemónicos Z80 que seguramente has leído y releído no aparece esta instrucción. Pues bien, resulta que DEFB es una pseudoinstrucción, es decir que al Z80 no le dice nada, pero al ensamblador le indica que los siguientes valores debe introducirlos directamente en memoria (DEFine Byte). Deberás pasar por algunas «experiencias» de este tipo antes de lograr el dominio completo de tu ensamblador, pero, si no desfalleces, en poco tiempo puedes conseguirlo. Cuenta con nosotros para cualquier otro problema que te surja.**

P ¿Cómo puedo crear números aleatorios en código máquina? ¿Podéis decirme cómo relacionar la forma de coordenadas X (0-255) e Y (0-175) con los archivos de pantalla (16384-22528)? Quiero saberlo para hacer un *fill* en C/M, no me aclaro con la forma de guardar la pantalla que tiene el Spectrum.

Quiero poner en contacto dos Spectrum a través de las conexiones del *cassette.* Para hacerlo emitiría por el canal 254 un 1 o un 0 y lo recibiría en el otro Spectrum por el 253. Lo que quiero saber es si hay interferencias, y si no las hubiese qué retardo tendría que poner entre señal y señal para poder transmitir sin errores.

Miguel Angel Olivar
Madrid

PROXIMO MES... PROXIMO MES... PROXIMO MES... PR

HELP Y BETA BASIC. Utilidades muy útiles.

• • • • • • • •

Cálculo de estructuras con Spectrum.

• • • • • • • •

Microdrive, ese desconocido.

• • • • • • • •

Graphi-QL, un programa de dibujo muy especial.

¡OH QUE MORENO VIENES! ¿POQUITO CREMA?

Preguntas y respuestas

R Para generar números aleatorios desde el código máquina podemos aprovecharnos del hecho de que la memoria RAM del ordenador, para que no se pierda, necesita ser «leída» con mucha frecuencia. Esta función la realiza la ULA constantemente, «refrescando» la memoria completa en pocas milésimas de segundo. El registro R lleva la cuenta de la zona de memoria que en ese momento se está «refrescando», por lo que si conseguimos el valor de este registro obtendremos un número difícilmente predecible (ya que varía constantemente y a gran velocidad). Esto lo podemos hacer gracias a una instrucción del Z80 (LD A,R) que hace precisamente eso: carga en el registro A el valor del registro de «refresco». Una vez hecho esto tendremos en el registro A un valor pseudoaleatorio que estará comprendido entre 0 y 127, las aplicaciones que queramos darle son ya otro problema...

En cuanto a tu segunda pregunta estamos totalmente de acuerdo contigo: ¡este sistema de almacenar la pantalla es verdaderamente endiablado! ¡Pero, oh maravilla de ROM! Disponemos de una rutina (PIXEL ADDRESS) ubicada en la dirección 8874 (22AA hex) cuyo propósito es aliviarnos de este problema: si la llamamos (CALL 8874) con el valor de la coordenada X en el registro B y el de Y en C, volverá dándonos su dirección en el archivo de pantalla en el par de registros HL y su posición dentro del byte en el registro A (¡ojo!, da la posición de izquierda a derecha, es decir, 0 para el bit 7 y 7 para el bit 0).

El método de comunicar dos Spectrum mediante EAR y MIC al que haces alusión es, en teoría, perfectamente practicable. No deben producirse ningún tipo de interferencias. El sistema, de todos modos, es muy delicado. Deberás ante todo idear un protocolo que ponga de acuerdo a los dos ordenadores; para ello has de tener perfectamente calculados los tiempos de ejecución de cada instrucción que utilices. También es posible que necesites crear buffers para almacenamiento temporal de datos. Para recibir de la conexión EAR deberás hacer IN del bit 6 del port 254 y OUT al bit 3 del mismo port para «excitar» la conexión MIC. Esperamos noticias tuyas (con el problema resuelto, por supuesto).

P Les mando esta carta con el fin de si serían tan amables de decirme cómo puedo introducir el programa RESET publicado en el número dos de su revista TODOSPECTRUM, ya que a mí que no tengo ni idea de código máquina o assembler me es imposible descifrar todas las cosas que pone en dicho programa.

Otro motivo de esta carta, y también relacionado con el código máquina, es el siguiente: yo no sé cómo introducir todas las sentencias que hay en los programas de este tipo y quería preguntarles cómo se introducen estas sentencias por medio de POKES; o sea que lo que quiero saber es por qué números se deben cambiar estas sentencias para ser introducidas.

Alberto Sánchez
Alcalá de Henares

R En la sección «Preguntas y Respuestas» del número seis publicamos un programa en BASIC, enviado por Aitor García, que cargaba esta rutina en una sentencia REM al principio del programa. Copiando este programa al pie de la letra no debes tener ningún problema para disfrutar de esta nueva función en tu ordenador.

En cuanto a tu segunda pregunta te recomendamos adquieras algún libro de iniciación al código máquina, donde puedas aprender todo lo referente a este lenguaje. De todas formas en el apéndice A del manual y junto al juego de caracteres encontrarás todos los mnemónicos del lenguaje ensamblador y sus correspondientes códigos en decimal y hexadecimal.

DIRECTOR: Simeón Cruz
COORDINADOR EDITORIAL: Emiliano Juárez
REDACCION: Juan Arencibia, Fernando García, José C. Tomás, Luis M. Brugarolas, Ricardo García, Santiago Gala
DISEÑO: Ricardo Segura

Editado por PUBLINFORMATICA, S. A.
Presidente: Fernando Bolín
Director Editorial: Norberto Gallego

Administración: INFODIS, S. A.
Gerente de Circulación y ventas: Luis Carrero
Producción: Miguel Onieva
Director de Marketing: Antonio González
Servicio al cliente: Julia González. Tel. 733 79 69
Administración: Miguel Atance y Antonio Torres
Jefe de Publicidad María José Martín
Dirección y redacción: Bravo Murillo, 377-5.º A. Tel. 733 74 13
Telex: 48877 OPZX e 28020 Madrid
Administración y Publicidad: Bravo Murillo, 377-3 E. Tels. 733 96 62/96
Publicidad Madrid: María Jose Martín
Publicidad Barcelona: María del Carmen Ríos, Olga Martorell. Pelayo, 12. Tel. (93) 318 02 89. 08001 Barcelona.
Depósito legal: M-29041-1984
Distribuye S.G.E.L. Avda. Valdelaparra, s/n. Alcobendas-Madrid.
Fotomecánica: Karmat, C/ Pantoja, 10. Madrid.
Fotocomposición: Artecomp.
Imprime: Héroes, C/ Torrelara, 8. Madrid.
Distribuidor en VENEZUELA, **SIPAM, S.A.** AVD. REPUBLICA DOMINICANA, EDIF. FELTREC – OFICINA 4B BOLEITA SUR CARACAS (VENEZUELA)
Esta publicación es miembro de la Asociación de Revistas de Información asociada a la Federación Internacional de Prensa Periódica, FIPP.

SUSCRIPCIONES:
Rogamos dirijan toda la correspondencia relacionada con suscripciones a:
TODOSPECTRUM
EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5.º
28002 MADRID
(Para todos los pagos reseñar solamente TODOSPECTRUM)
Para la compra de ejemplares atrasados diríjanse a la propia editorial
TODOSPECTRUM
C/ Bravo Murillo, 377-5.º A
Tel. 733 74 13 - 28020 MADRID

Si deseas colaborar en TODOSPECTRUM remite tus artículos o programas a Bravo Murillo 377, 5.º A. 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.

SEIKOSHA SP-800

El fruto de la Investigación



La nueva impresora de **SEIKOSHA SP-800,** con un ordenador personal puede escribir ***96 combinaciones de letra diferentes,*** desde 96 caracteres por segundo a 20 con muy alta calidad de letra, además es gráfica en alta densidad.

Su precio es de 69.900 R con ***introductor automático hoja a hoja.***

Con un pequeño ordenador personal, un procesador de textos puede costar alrededor de cien mil pesetas.

Infórmese y comprenderá por qué **las máquinas de escribir** tienen demasiados años.

Nuestra calidad es "SEIKO";
nuestros precios, únicos
Si desea más información,
consulte con nuestro distribuidor
más cercano, llame o escriba a:

ESTOS SON NUESTROS MODELOS:

MODELO	VELOCIDAD	COLUMNAS	TIPOS DE LETRA	P.V.P.R. INTERFACE PARALELO
GP-50S LA DEL SPECTRUM	40 cps	32	-	19.900
GP-50 LA PEQUEñA	40 cps	46	2	25.900
GP-500 LA ECONOMICA	50 cps	80	2	47.900
GP-700 LA DE COLOR	50 cps	80-106	3	69.900
SP-800 LA PERFECCION	96 cps	80-137	20	69.900
BP-5200 LA DE OFICINA	200 cps	136-272	10	199.900
BP-5420 LA MAS RAPIDA	420 cps	136-272	10	299.900

* Los precios indicados son los recomendados para conexion tipo paralelo Centronics, para otro tipo de conexion, sufren un ligero incremento.

DiRAC

DIRECCION COMERCIAL:
Av. Blasco Ibañez, 114-116
46022 VALENCIA
Tel.(96) 372 88 89
Télex 62220
DIRECCION COMERCIAL EN CATALUÑA:
C/Muntaner, 60-2-4Pta
08011 BARCELONA
Tel.(93) 323 32 19

Este pie de página ha sido realizado integramente con la nueva impresora:

SEIKOSHA SP-800